

Spatial and Temporal Interpolation of Multi-View Video

Tobias Gurdan

Bachelor's Thesis in Computer Science

Supervisor: Prof. Dr. Daniel Cremers

Advisors: Martin Oswald, Daniel Gurdan

Outline

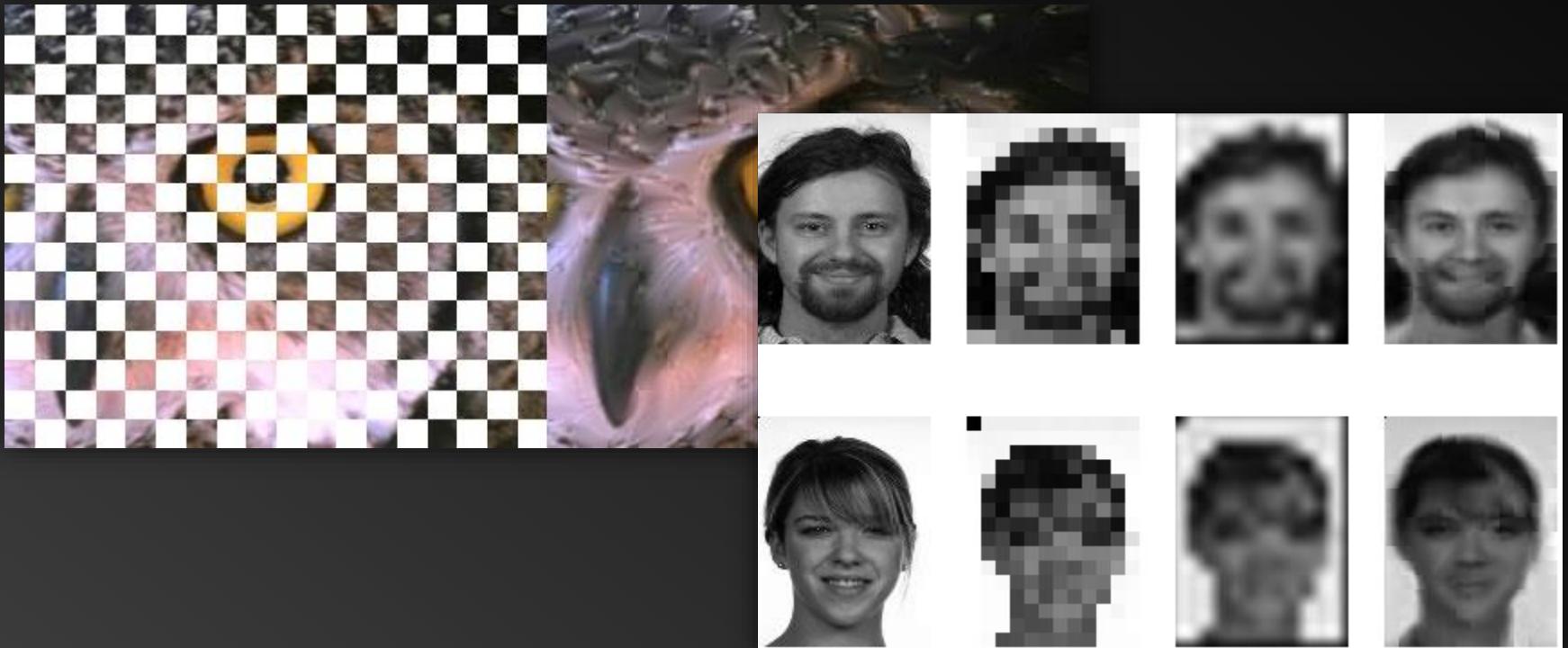
Outline

1. Introduction
2. Temporal Interpolation
 1. Dense Matching
 2. Image Warping
3. Spatial Interpolation
 1. Features
 2. Sparse Matcher
 3. Triangulation
 4. View Morphing
4. Discussion
5. Results

Introduction

Introduction

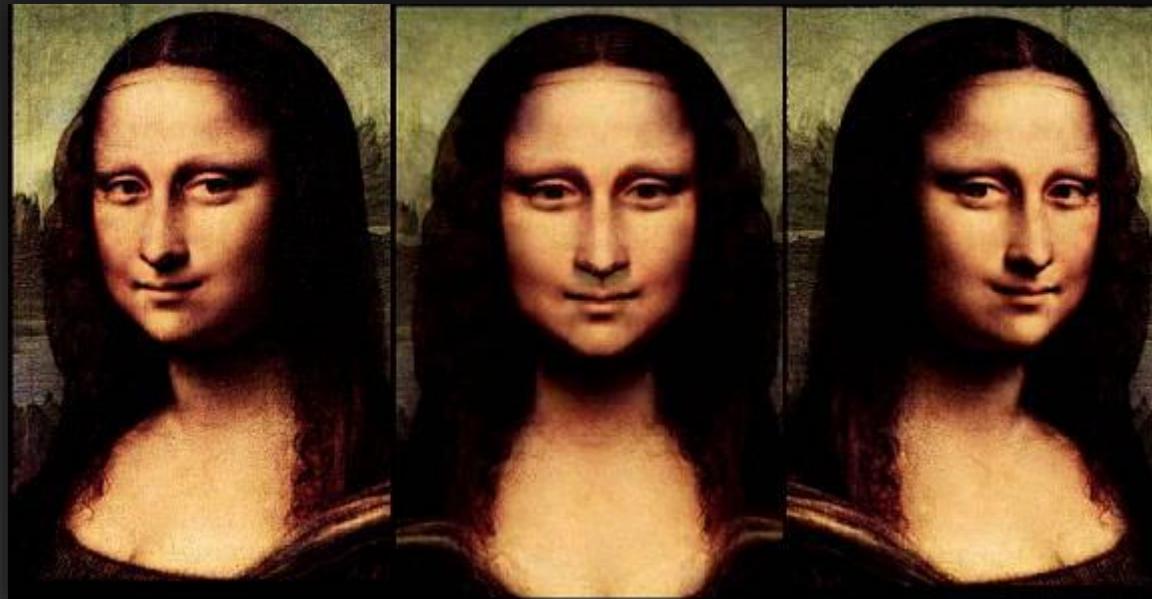
- What is image interpolation?



David Tschumperlé, <https://tschumperle.users.greyc.fr/publications.php> ,
Digital Multi Media Design, http://dmmd.net/main_wp/research/interpolation/

Introduction

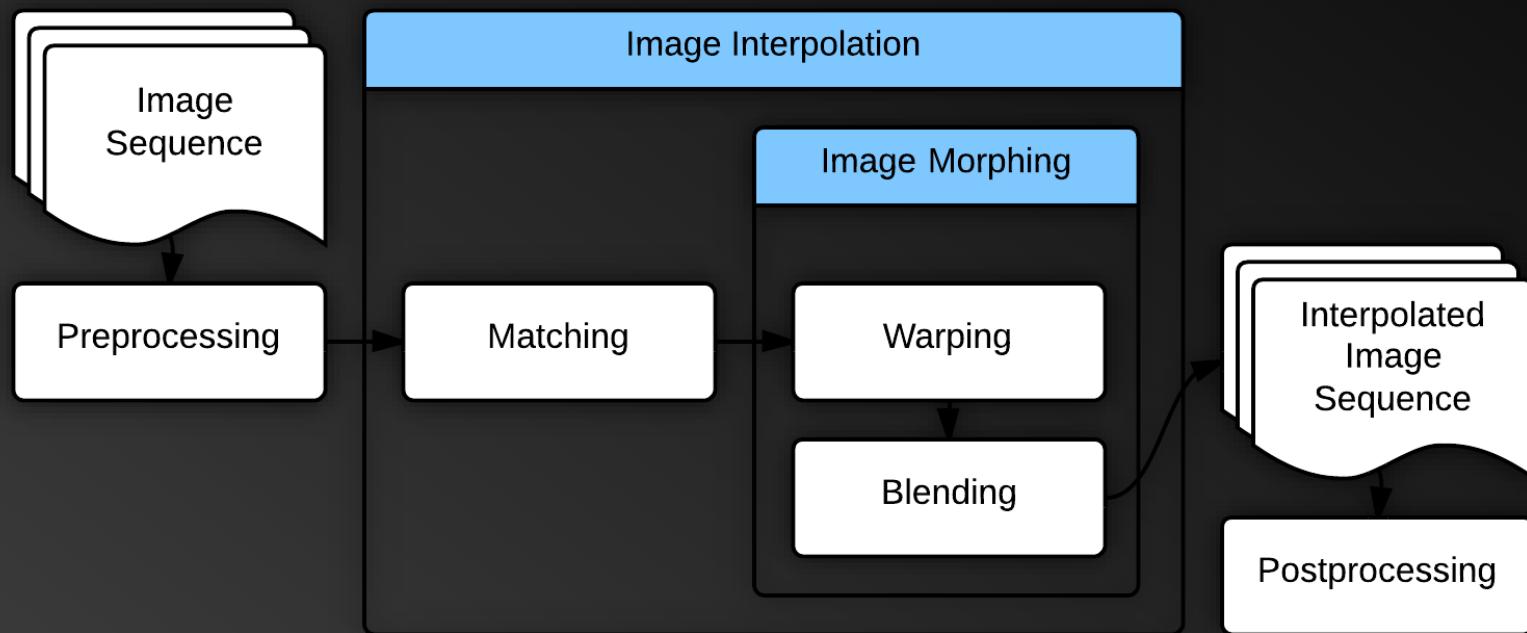
- What is image interpolation?



Seitz and Dyer [SD96]

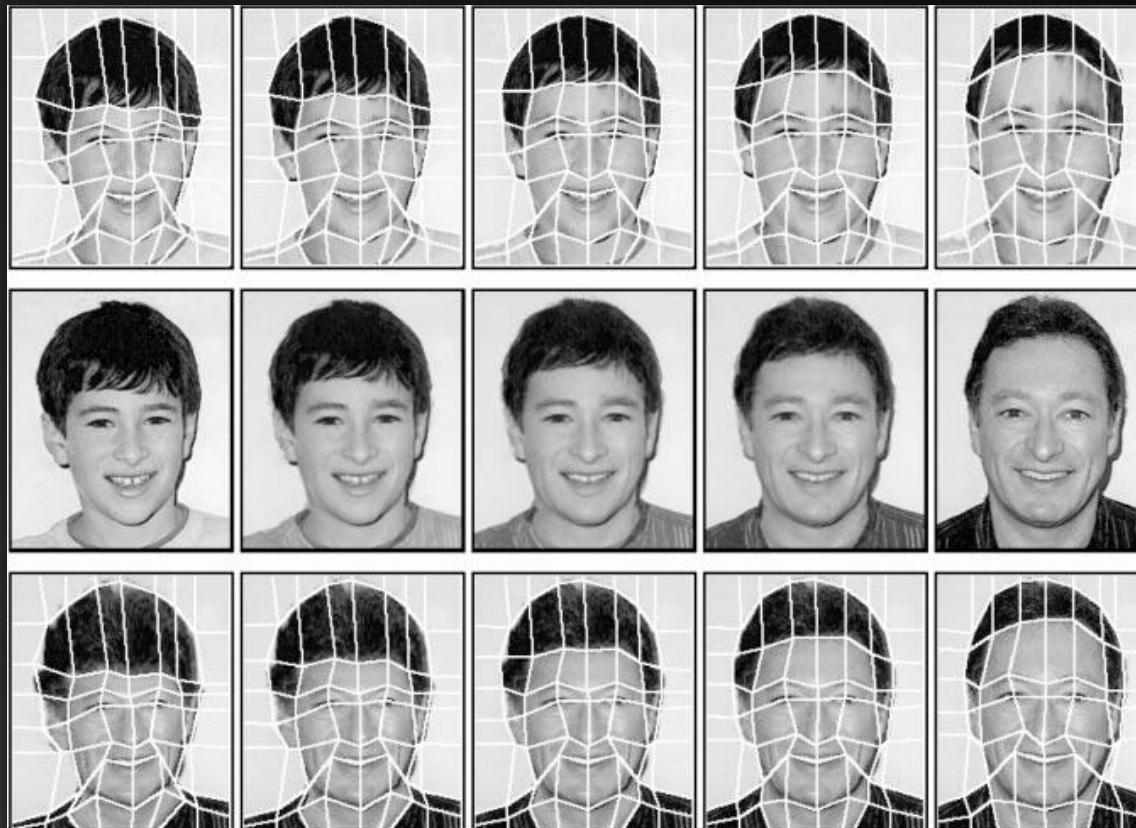
Introduction

- Pipeline



Introduction

- Example: Face Morphing [Wol98]



Introduction

- Related work

Introduction

- Related work
 - Beier and Neeley [BN92]

Introduction

- Related work



Introduction

- Related work
 - Beier and Neeley [BN92]
 - Seitz and Dyer [SD95, SD96]
 - Zitnick et al. [ZKU+04]

Introduction

- Related work



Introduction

- Related work
 - Beier and Neeley [BN92]
 - Seitz and Dyer [SD95, SD96]
 - Zitnick et al. [ZKU+04]
 - Lipski et al. [LLBM09]

Introduction

- Related work



**Our approach
(view interpolation)**

**Our approach
(view and time
interpolation)**

Introduction

- Related work
 - Beier and Neeley [BN92]
 - Seitz and Dyer [SD95, SD96]
 - Zitnick et al. [ZKU+04]
 - Lipski et al. [LLBM09]
 - Inamato and Saito [IS05], Replay Technologies Inc.

Introduction

- Related work



freeD™ technology at New York Yankee Stadium

Introduction

- Goal of this thesis



Introduction

- Goal of this thesis
 - Spatial and temporal interpolation
 - Easy to use
 - Autonomous
 - No restriction on setup



Temporal Interpolation

1) Dense Matching

Dense Matching

- Optical Flows
 - Dual TV-L1
 - Farneback
 - SimpleFlow
- Block Matching
 - Simple Block Matching
 - Semi Global Block Matching
- Manual Matching
 - Cross-Correlation along Scanlines

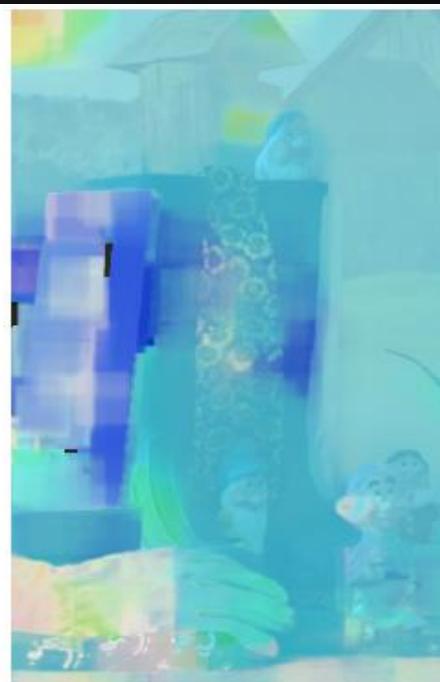
Dense Matching



(a) Dual TV-L1



(b) Farneback



(c) Simple Flow



(d) Semi-Global Block Matching

Temporal Interpolation

2) Image Warping

Image Warping

- Transform source image to target image

Image Warping

- Transform source image to target image
- Forward Mapping
 - Iterate over source image

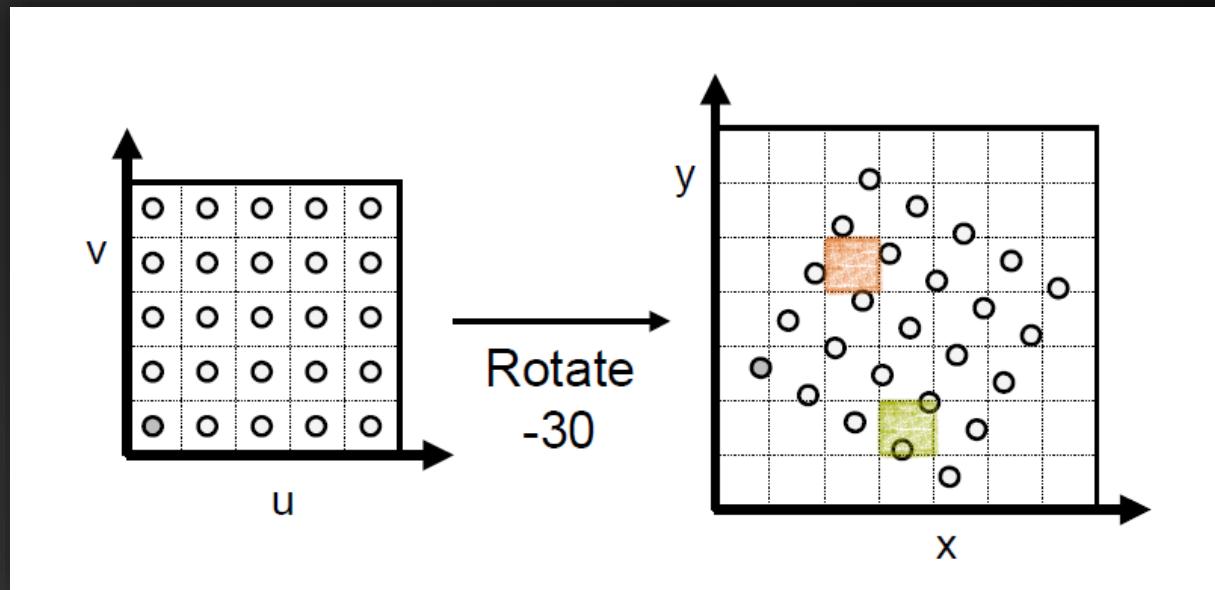


Image Warping

- Transform source image to target image
- Forward Mapping
 - Iterate over source image
 - Results in folds and holes

Image Warping

- Transform source image to target image
- Forward Mapping
- Backward Mapping

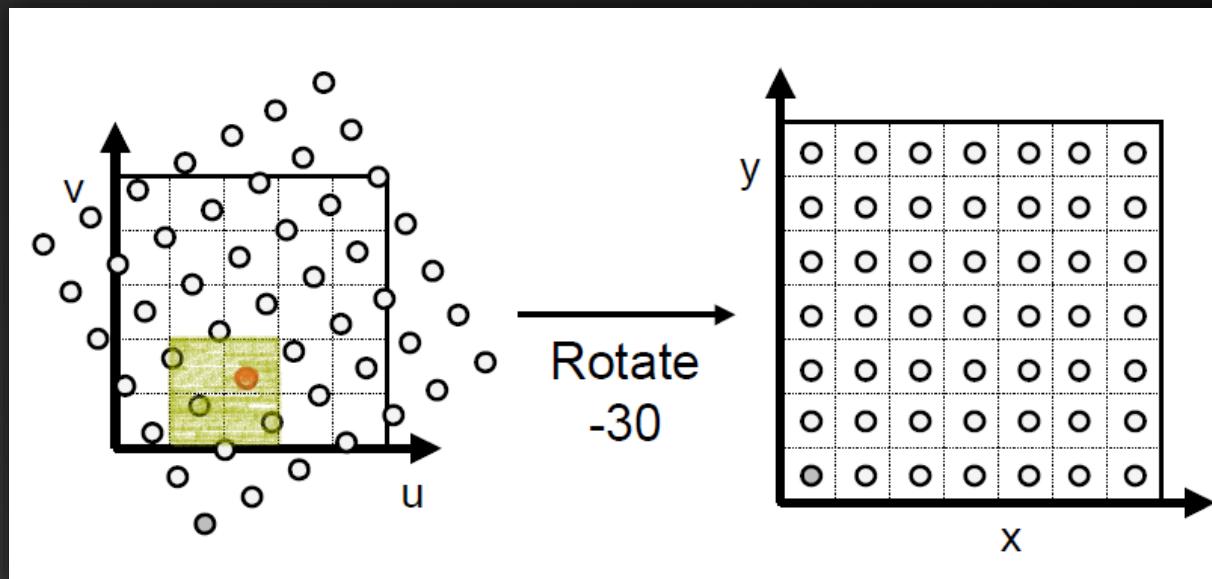


Image Warping

- Transform source image to target image
- Forward Mapping
- Backward Mapping
 - Use flow from target to source as inverse flow for source image
 - Gradually add displacements to original positions

Given flow \mathcal{F} from source image \mathcal{I}_1 to target image \mathcal{I}_2 . \mathcal{I}_1 can then gradually be warped towards \mathcal{I}_2 according to

$$\mathcal{I}(x, y, t) = \mathcal{I}_1(x + t \cdot \mathcal{F}_x^{-1}(x, y), y + t \cdot \mathcal{F}_y^{-1}(x, y)) , \quad t \in [0, 1] \quad (2.4)$$

In particular $\mathcal{I}(x, y, 0) = \mathcal{I}_1(x, y)$ and for perfect flows $\mathcal{I}(x, y, 1) = \mathcal{I}_2(x, y) \quad \forall x, y \in \Omega$.

Image Warping

- Transform source image to target image
- Forward Mapping
- Backward Mapping
 - Use flow from target to source as inverse flow for source image
 - Gradually add displacements to original positions
 - Implemented in fragment shader

Demo: Biker 1 & 2

Spatial Interpolation

Spatial Interpolation

1) Features

Features

- What is a feature?
 - Regions in an image, that satisfy certain criteria
-
- **Local** - Occupies a small area, thus being robust to clutter and occlusions
 - **Invariant** - Can be found across multiple images, regardless of geometric and photometric changes
 - **Distinctive** - Can be uniquely described
 - **Robust** - Noise, blur, quantization etc. do not destroy the feature description
 - **Efficient** - Noise, blur, quantization etc. do not destroy the feature description

Features

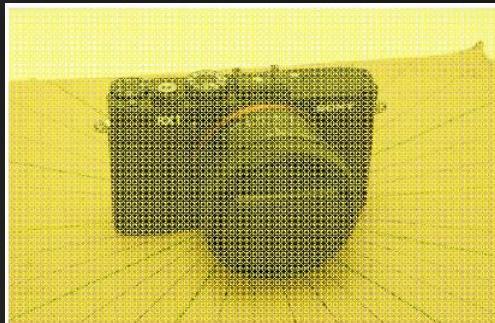
- What is a feature?
- Regions in an image, that satisfy certain criteria
- Find ‘good’ sets of features

- **Distribution** - Not too close to get good meshes, not too far apart which might lead to missed regions
- **Coverage** - Include most to all significant points, e.g. object outlines, corners, foreground / background

TOLEDO \ BACKPACK

Features

- Feature Detectors



(a) Dense features



(b) FAST features



(c) GFTT features



(d) ORB features



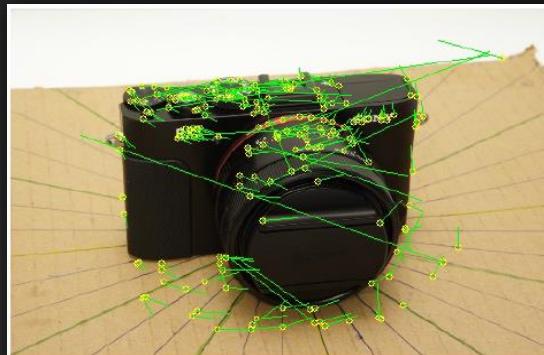
(e) SIFT features



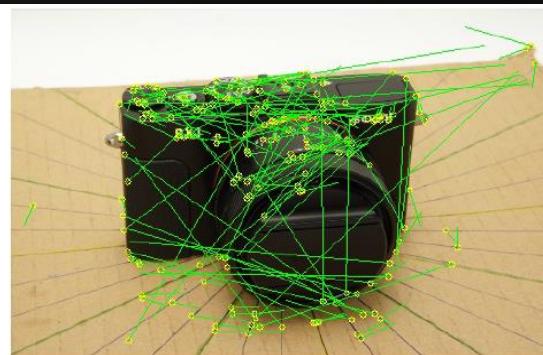
(f) SURF features

Features

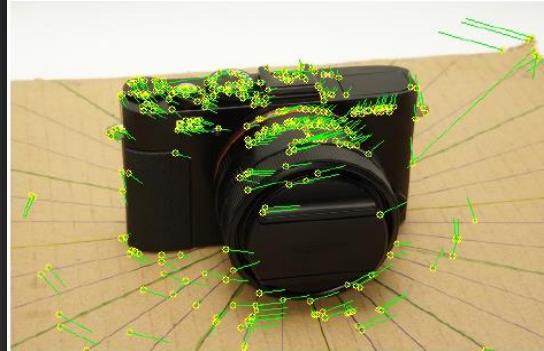
- Feature Descriptors



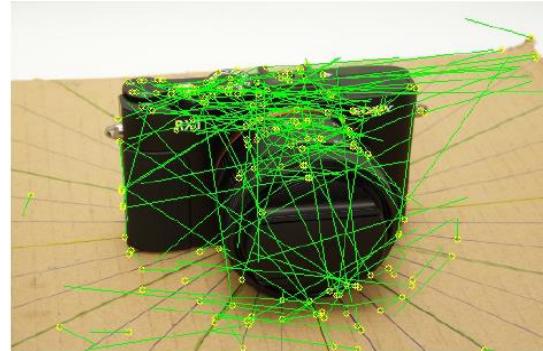
(a) BRIEF descriptor



(b) BRISK descriptor



(c) SIFT descriptor



(d) SURF descriptor

Spatial Interpolation

2) Matcher

Matcher

- Robust Matcher

Matcher

- Robust Matcher
 - Source and target image



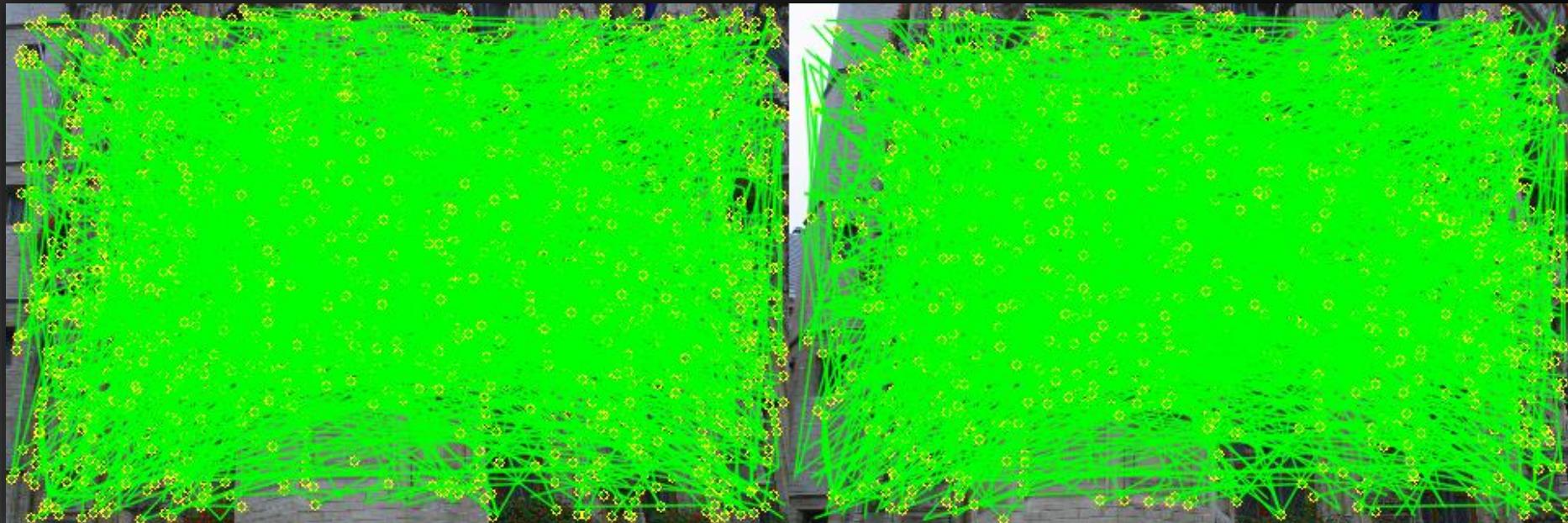
Matcher

- Robust Matcher
 - SIFT features



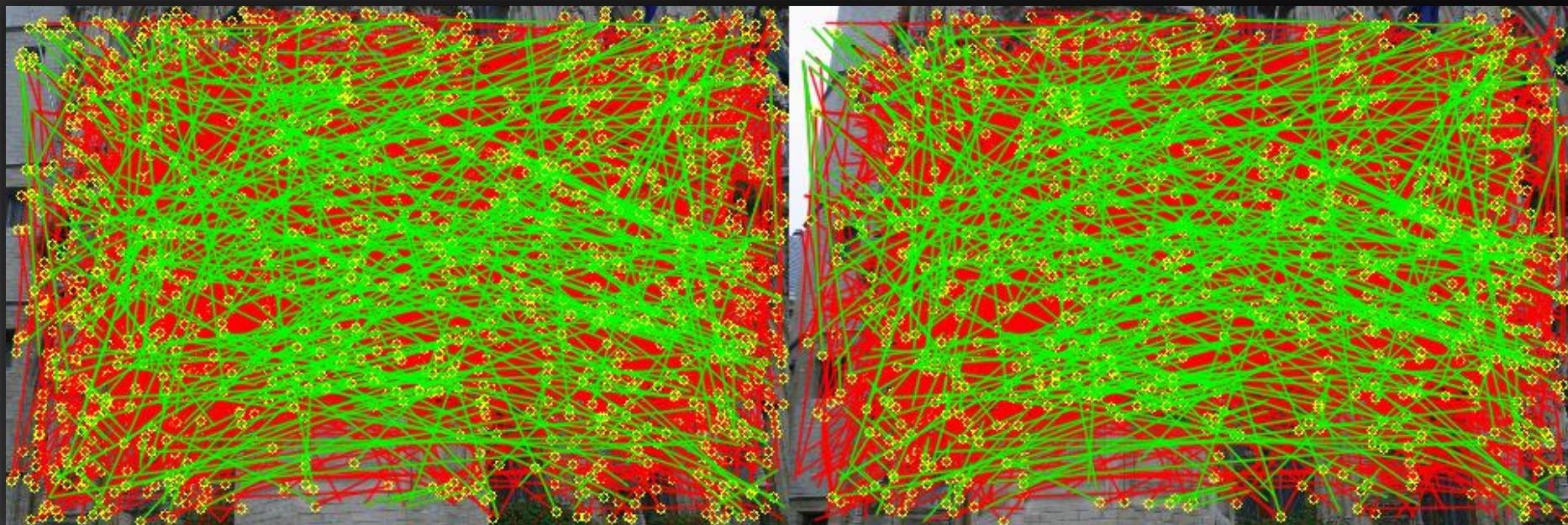
Matcher

- Robust Matcher
 - SIFT descriptor matching



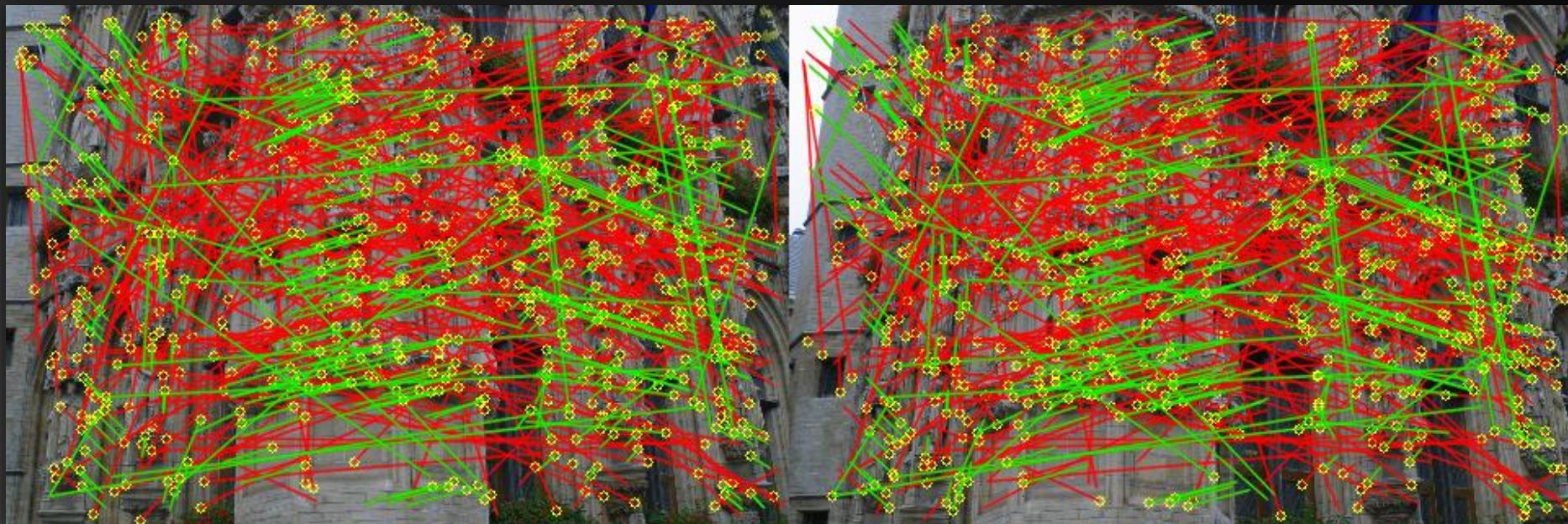
Matcher

- Robust Matcher
 - Symmetric matches



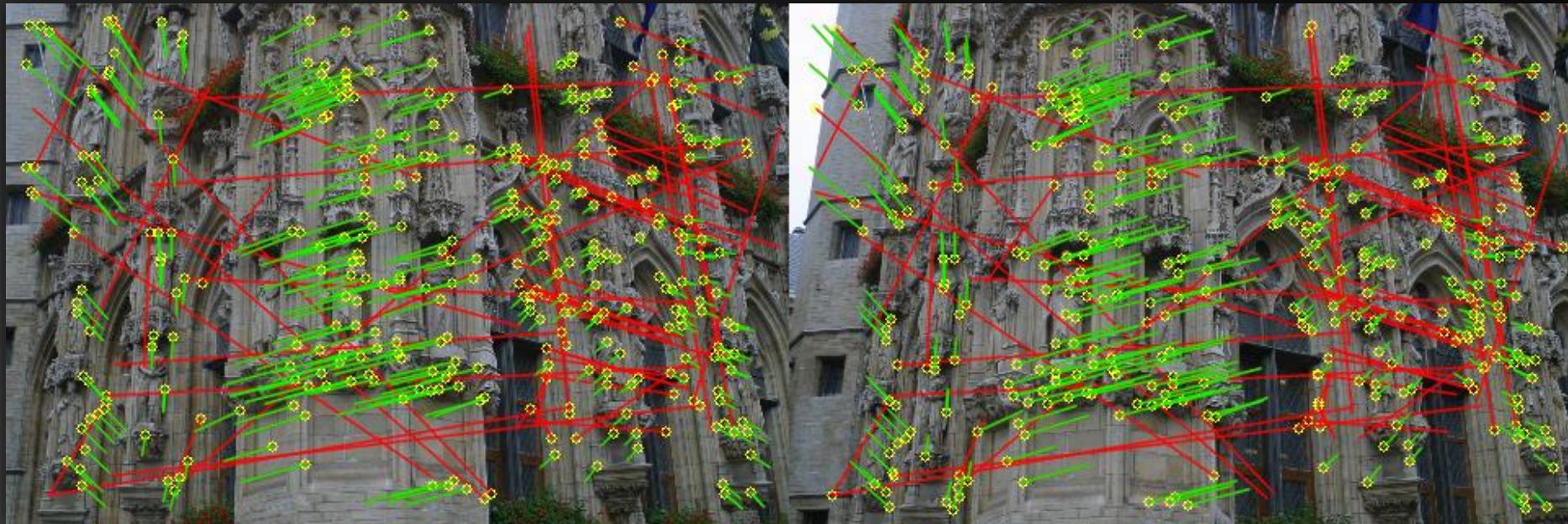
Matcher

- Robust Matcher
 - Thresholded matches



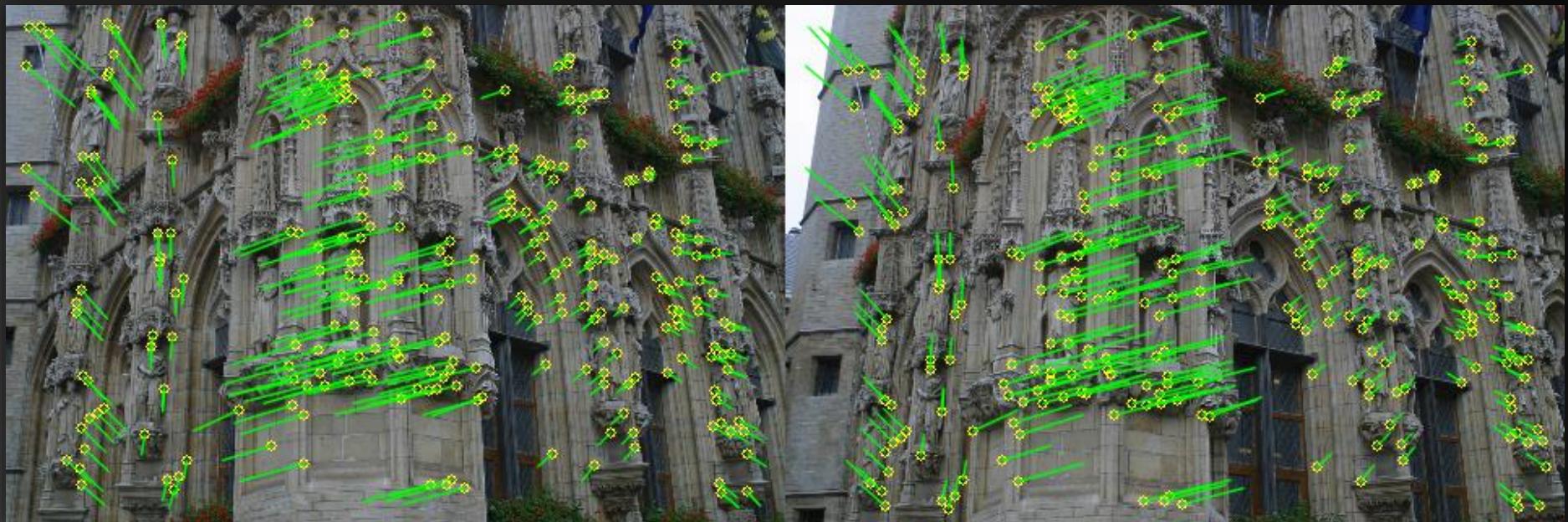
Matcher

- Robust Matcher
 - RANSAC filtering



Matcher

- Robust Matcher
 - Final matching



Matcher

- Robust Matcher
 - Works good on well textured scenes
 - SIFT features are robust but often not suitable for warping
 - Bruteforce matching as major weak point

Matcher

- Epipolar Guided Matcher
 - Constrain search space for matching
 - Use epipolar geometry and epipolar constraint

The epipolar geometry is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose. The fundamental matrix \mathbf{F} encapsulates this intrinsic geometry. It is a 3×3 matrix of rank 2. If a point in 3-space \mathbf{p} is imaged as \mathbf{x}_1 in the first view, and \mathbf{x}_2 in the second, then the image points satisfy the relation $\mathbf{x}_2 \mathbf{F} \mathbf{x}_1 = \mathbf{0}$.

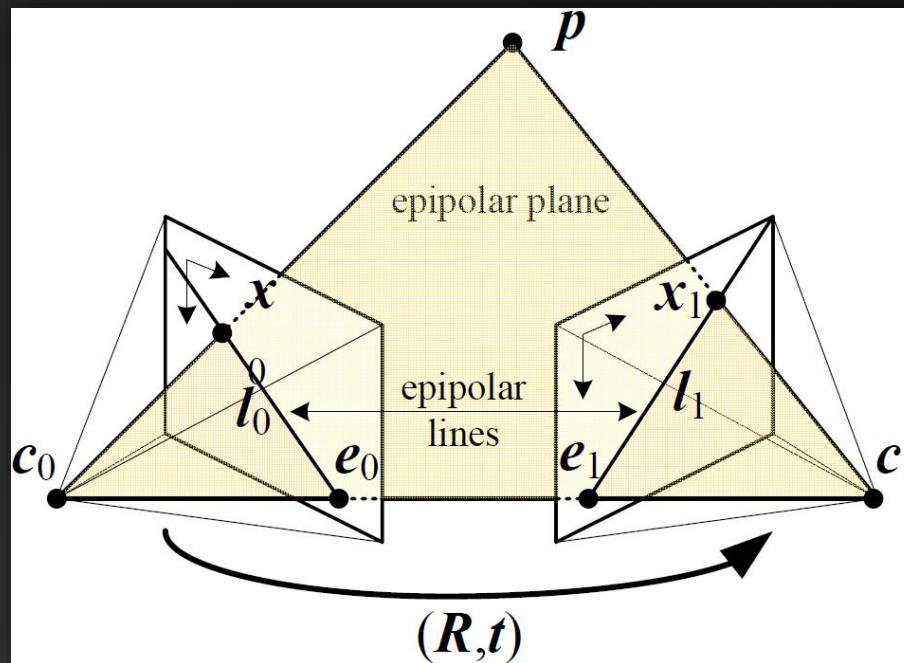
[HZ03, chap. 9, adjusted variables]

[HZ03, chap. 9, adjusted variables]

$$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0$$

Matcher

- Epipolar Guided Matcher
 - Constrain search space for matching
 - Use epipolar geometry and epipolar constraint



Matcher

- Epipolar Guided Matcher
 - Source and target image



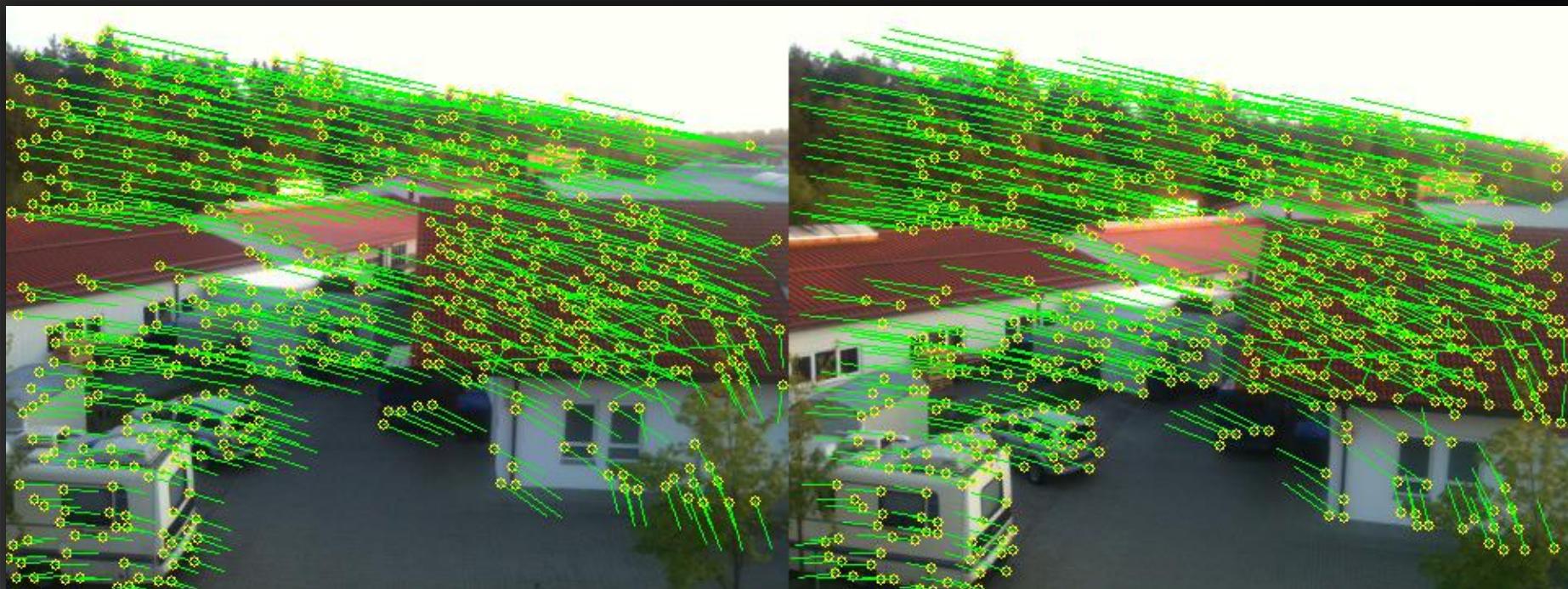
Matcher

- Epipolar Guided Matcher
 - Robust Matcher result



Matcher

- Epipolar Guided Matcher
 - Guided matching using epipolar constraint

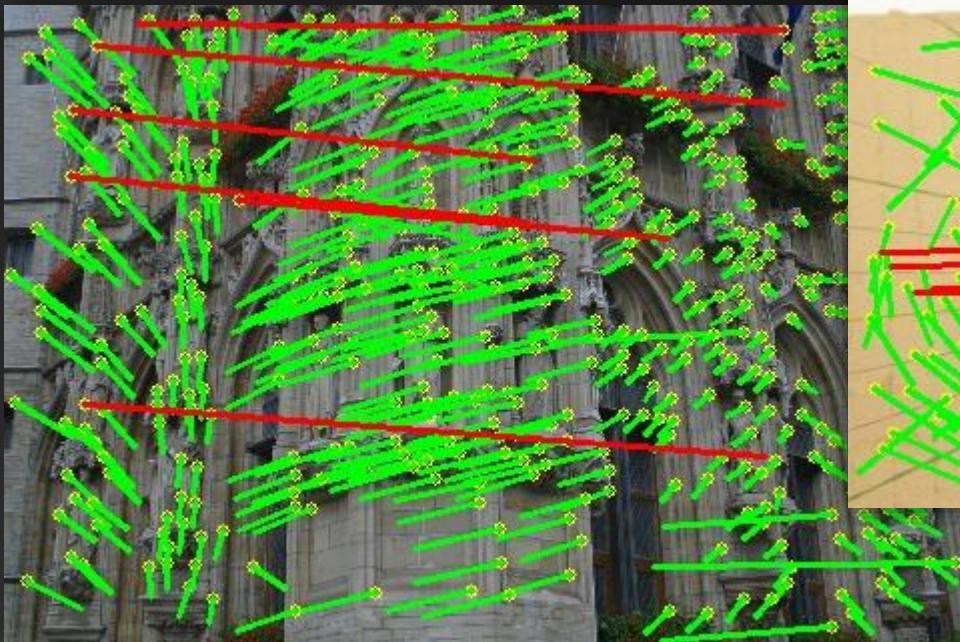


Matcher

- Epipolar Guided Matcher
 - Playground Filter
 - Global filtering
 - Partition matches into three clusters
 - Discard cluster with largest displacement ...
 - ... If the distance is above a certain threshold

Matcher

- Epipolar Guided Matcher
 - Playground Filter

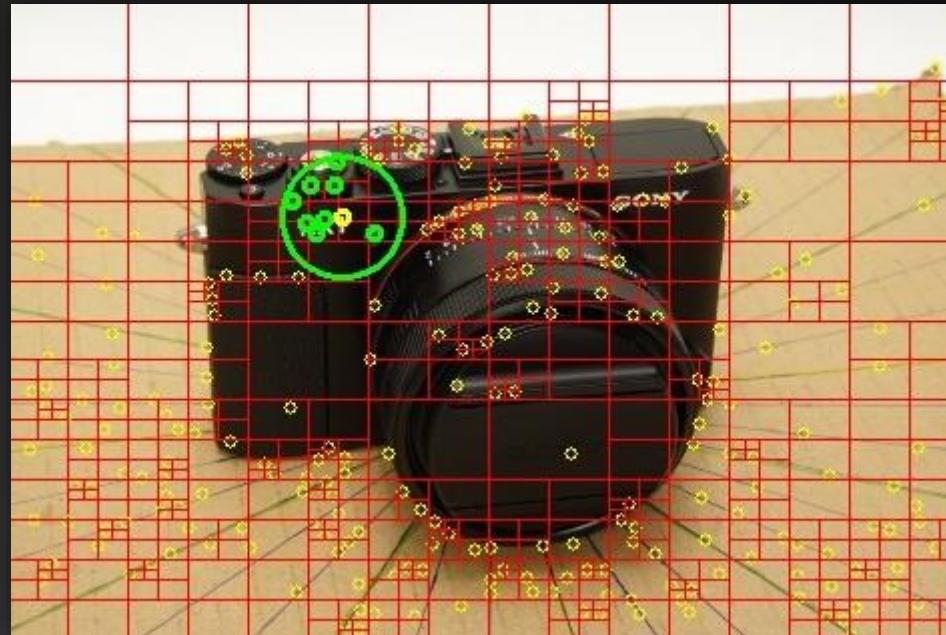


Matcher

- Epipolar Guided Matcher
 - Council Filter
 - Local filter
 - Look at small neighbourhoods
 - Each keypoint in that region votes for a candidate
 - Accept candidate as inlier only if it receives enough votes

Matcher

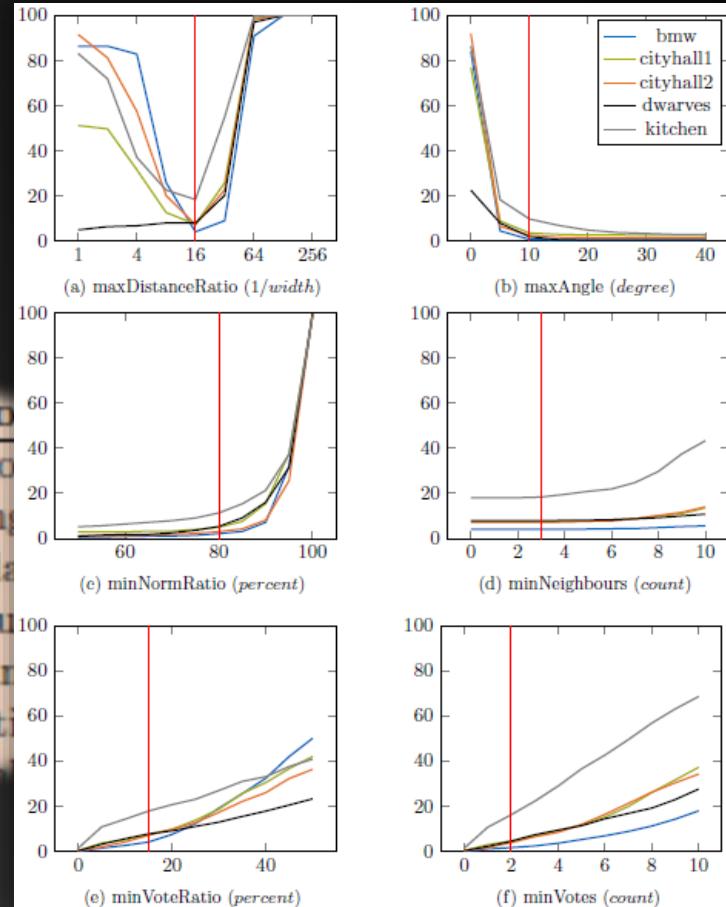
- Epipolar Guided Matcher
 - Council Filter



Matcher

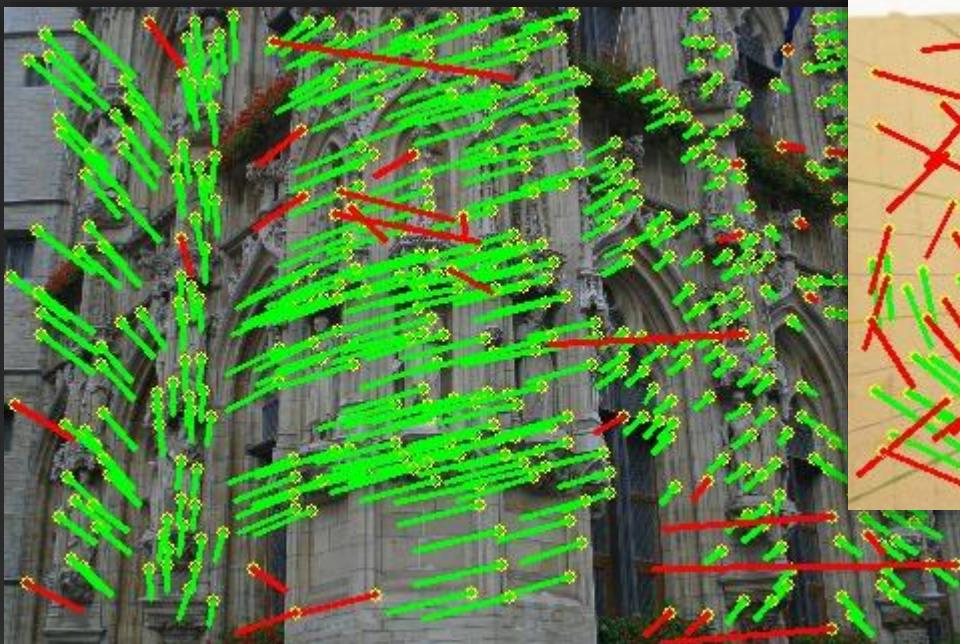
- Epipolar Guided Matcher
 - Council Filter

Parameter	Value	Explanation
maxDistanceRatio	1/16	neighbourhood size
maxAngle	10	maximal angle
minNormRatio	0.8	minimal relative ratio
minNeighbours	3	minimal count
minVotes	2	minimal number of votes
VoteRatio	0.15	minimal ratio
Pass	false	accept as is



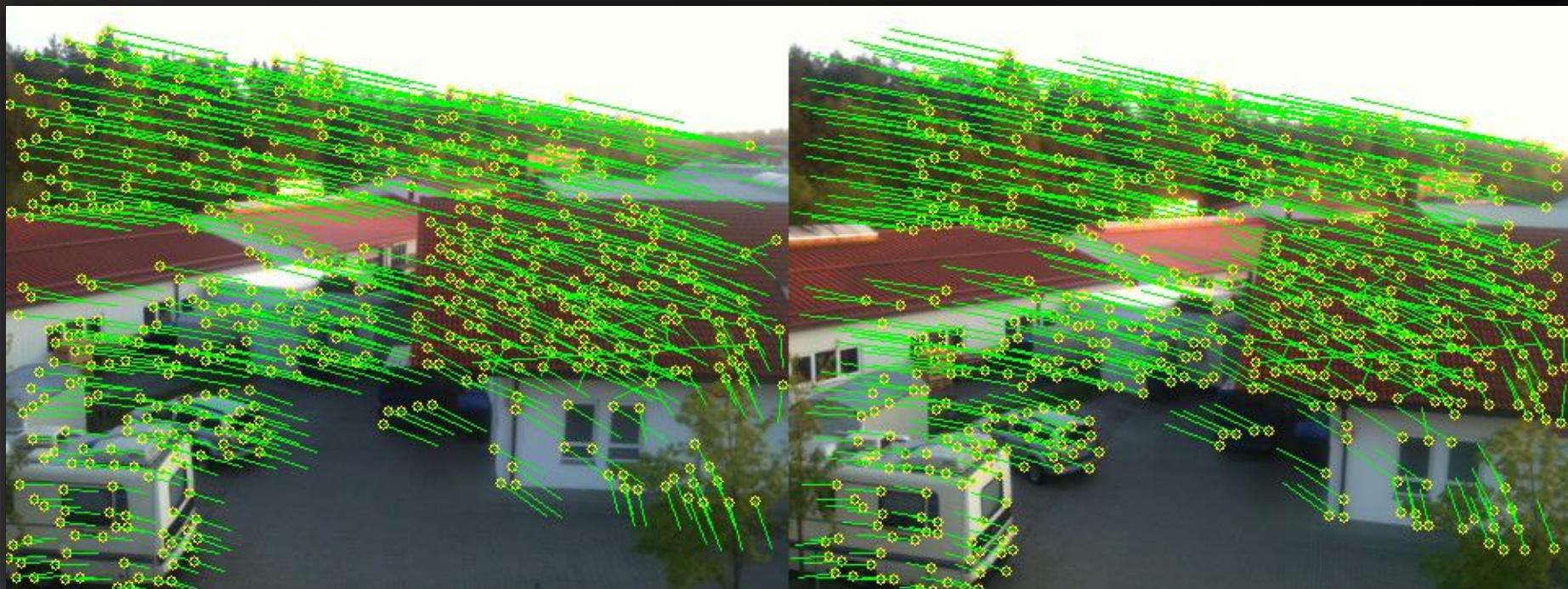
Matcher

- Epipolar Guided Matcher
 - Council Filter



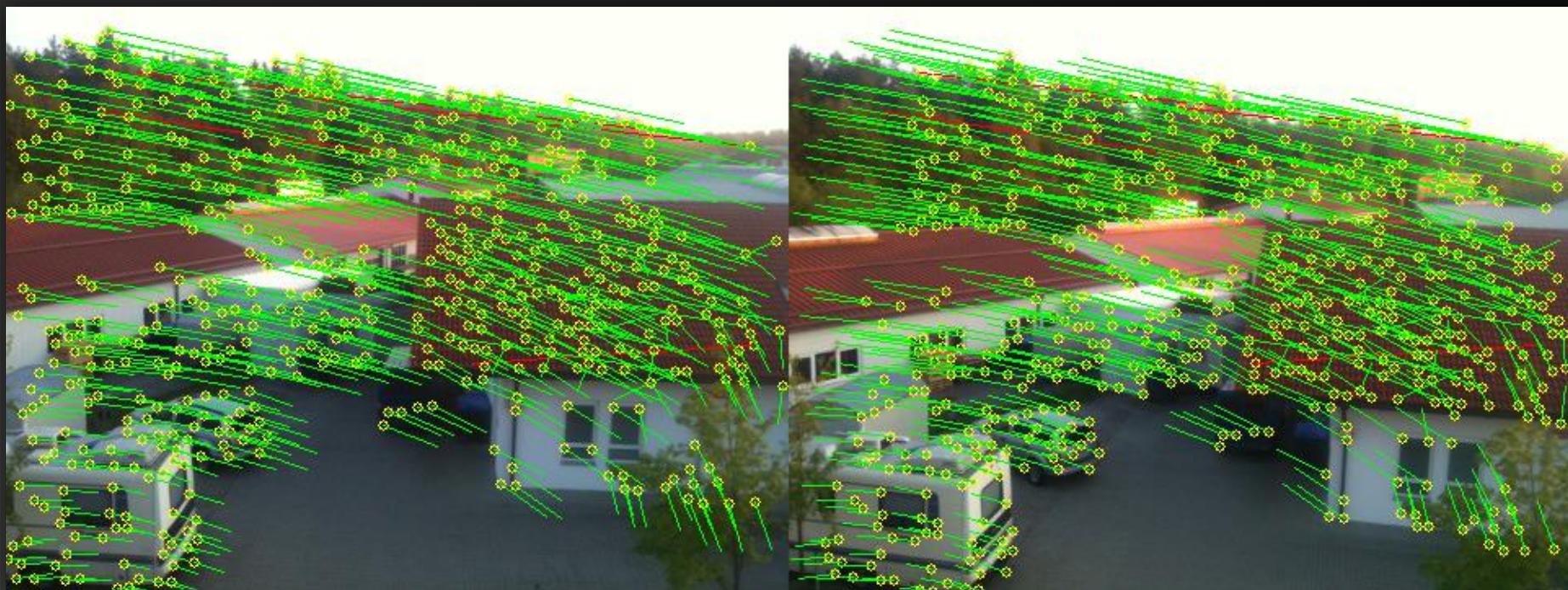
Matcher

- Epipolar Guided Matcher
 - Guided matching using epipolar constraint



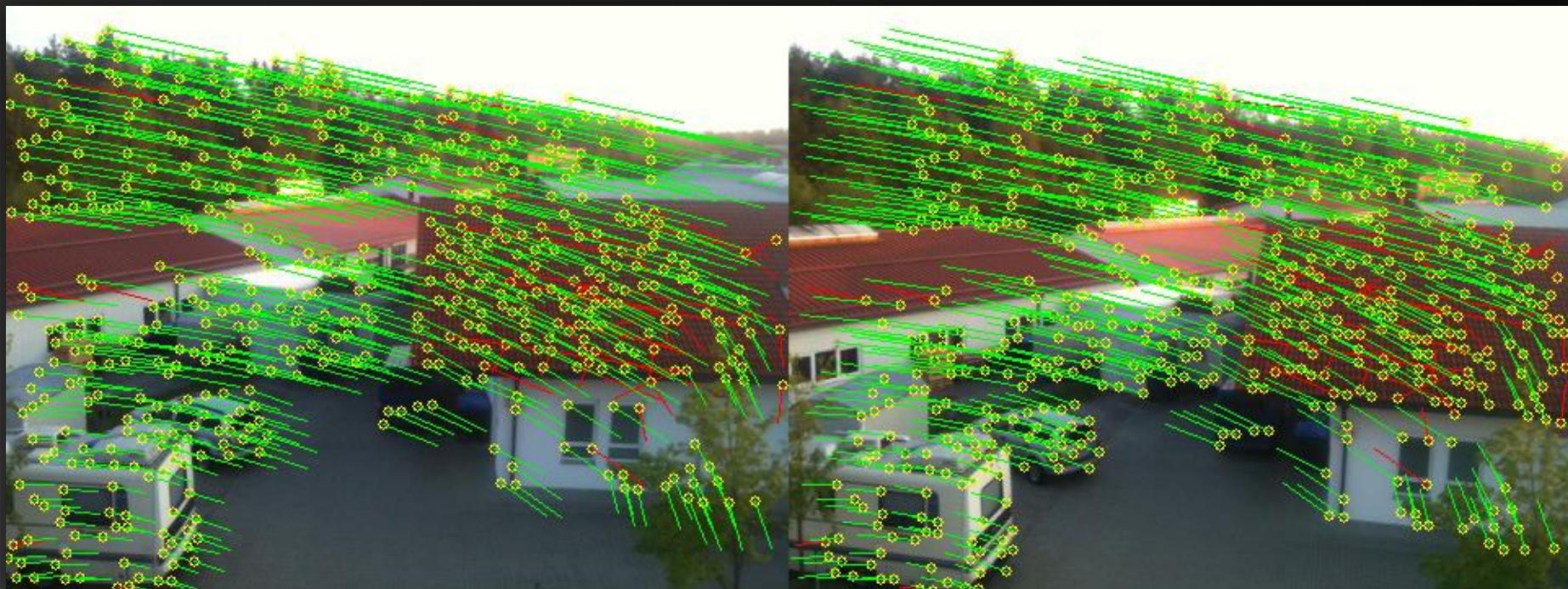
Matcher

- Epipolar Guided Matcher
 - Apply Playground Filter



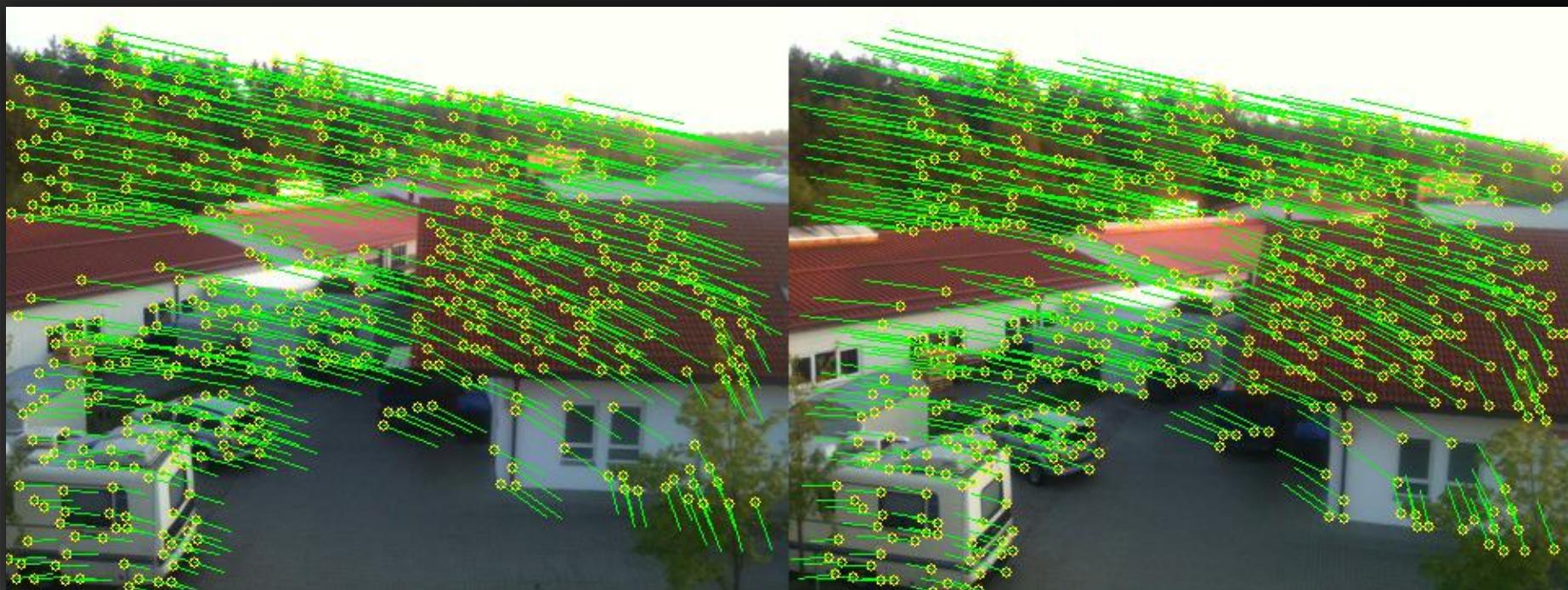
Matcher

- Epipolar Guided Matcher
 - Apply Council Filter



Matcher

- Epipolar Guided Matcher
 - Final matching

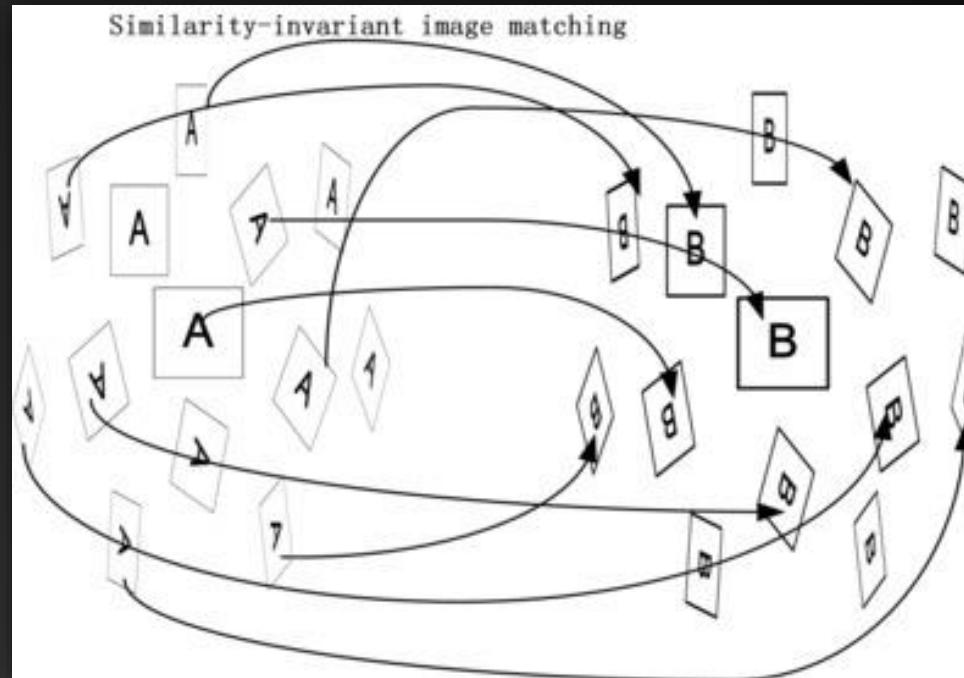


Matcher

- Epipolar Guided Matcher
 - Yields good results for most examples
 - Still fails to capture extreme scene motions ...
 - ... and to match low textured images

Matcher

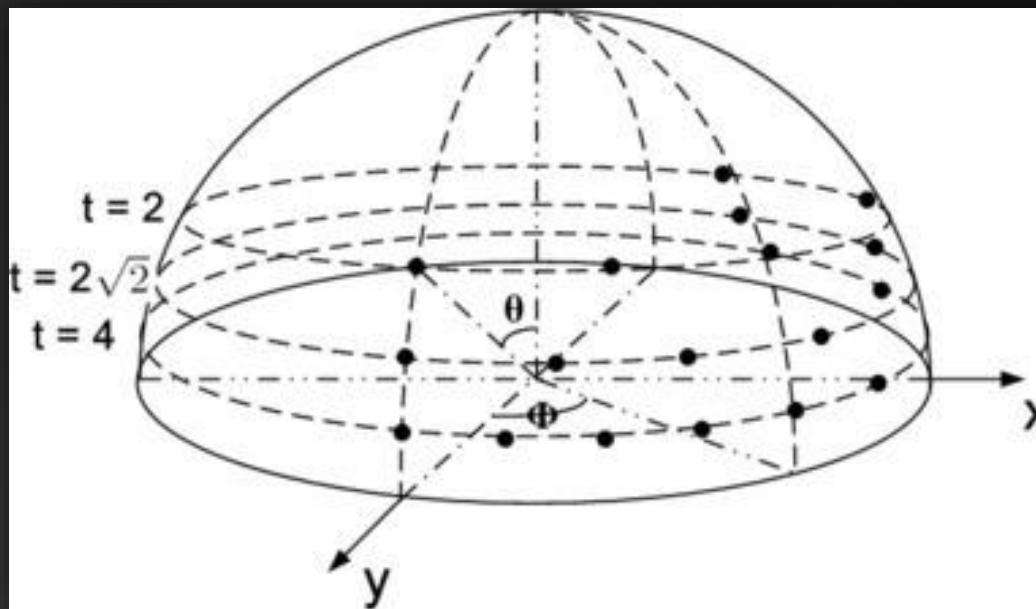
- ASIFT Matcher
 - Extends SIFT descriptor by missing affine parameters



Morel and Yu [MY09]

Matcher

- ASIFT Matcher
 - Extends SIFT descriptor by missing affine parameters
 - Sample parameter space and simulate affine transformations



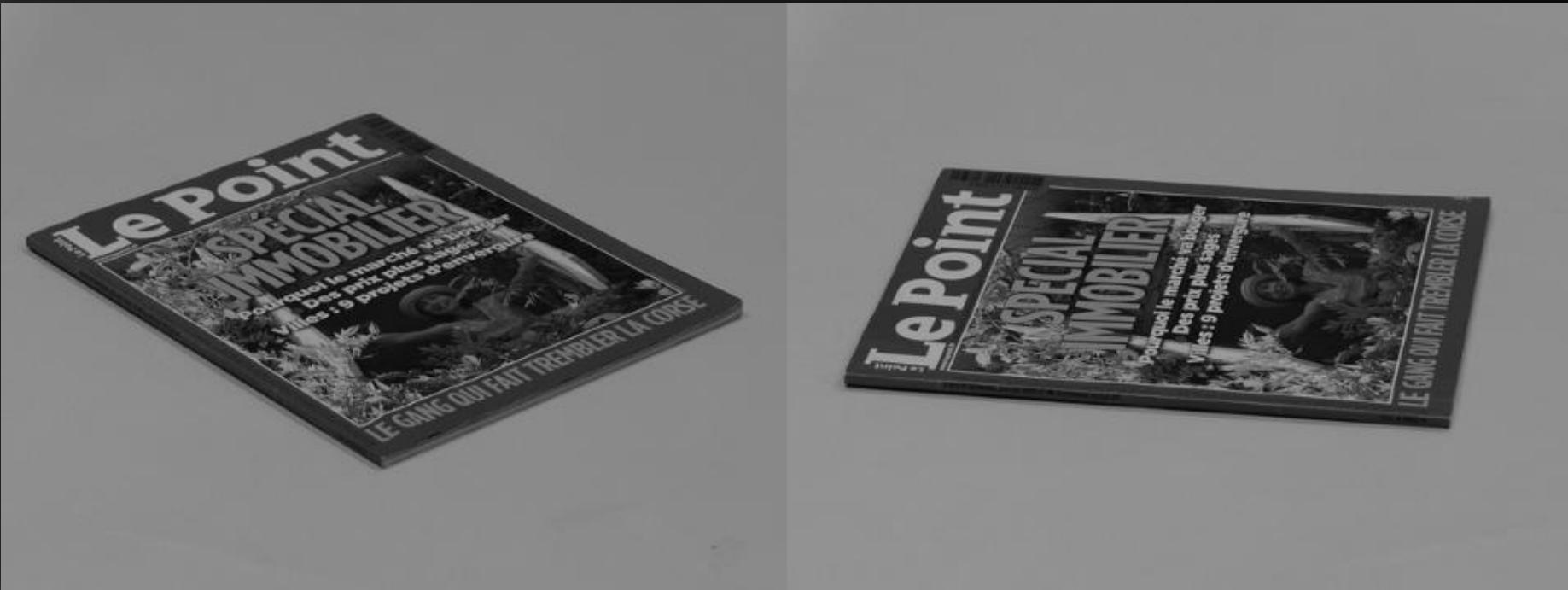
Morel and Yu [MY09]

Matcher

- ASIFT Matcher
 - Extends SIFT descriptor by missing affine parameters
 - Sample parameter space and simulate affine transformations
 - Filter, filter, filter (identical, ambiguous, inlier matches)

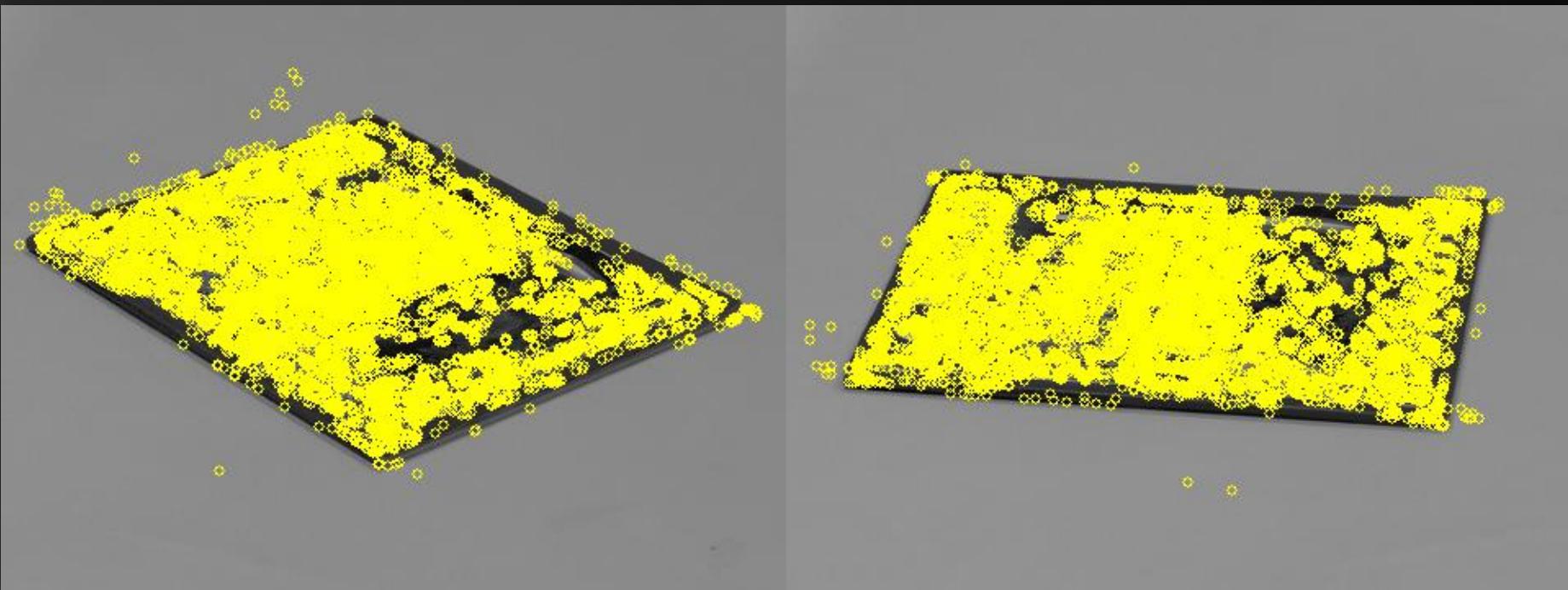
Matcher

- ASIFT Matcher
 - Source and target image



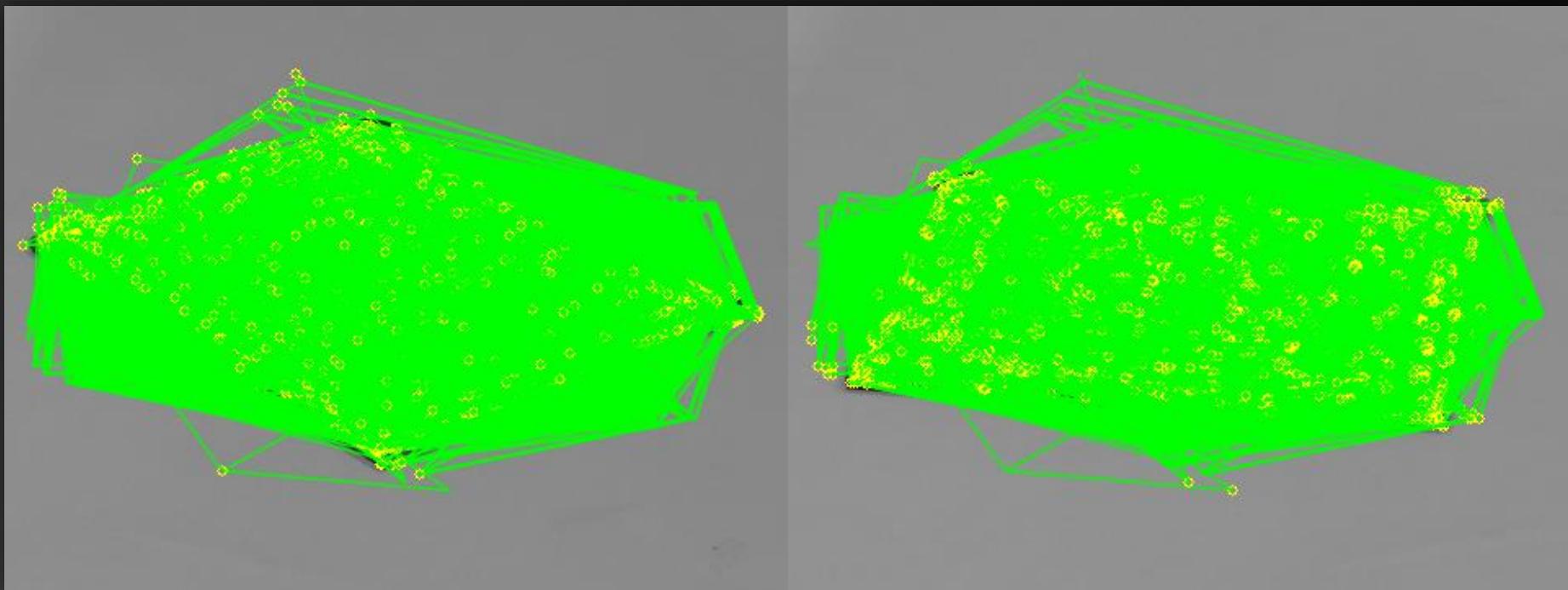
Matcher

- ASIFT Matcher
 - Keypoints of all simulations



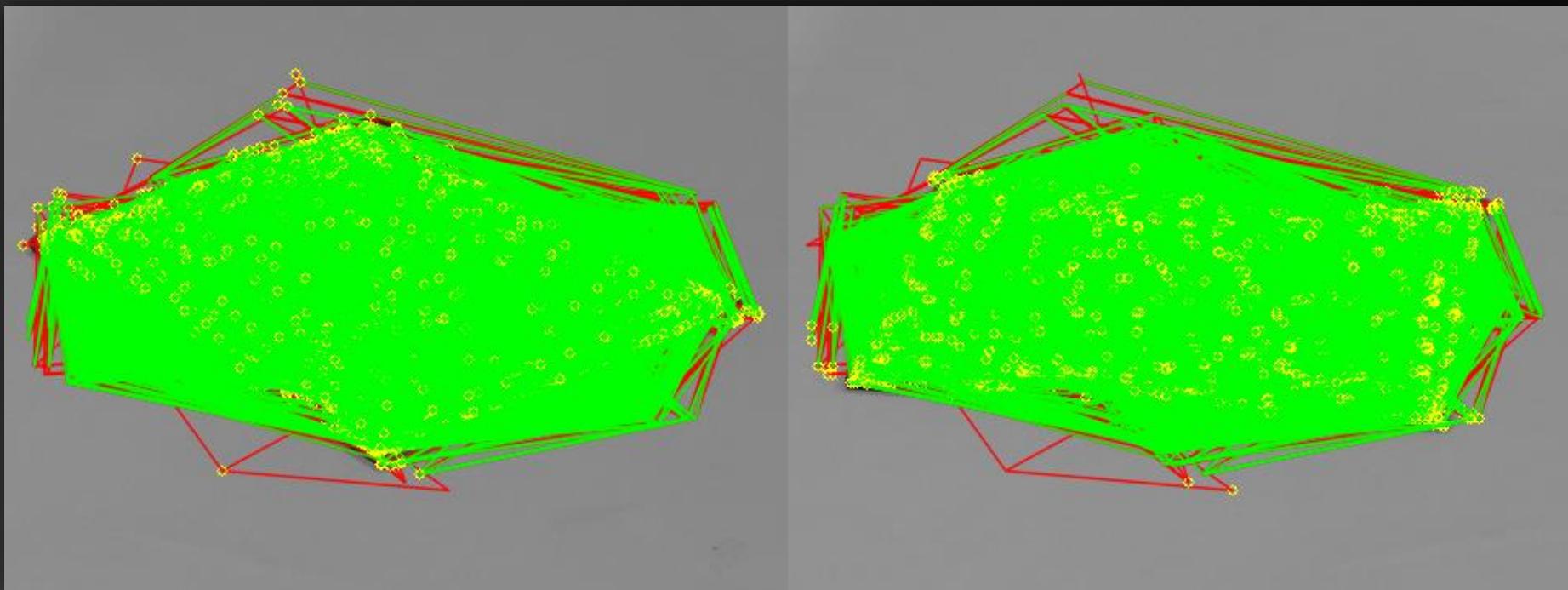
Matcher

- ASIFT Matcher
 - Matching for all parameter combinations



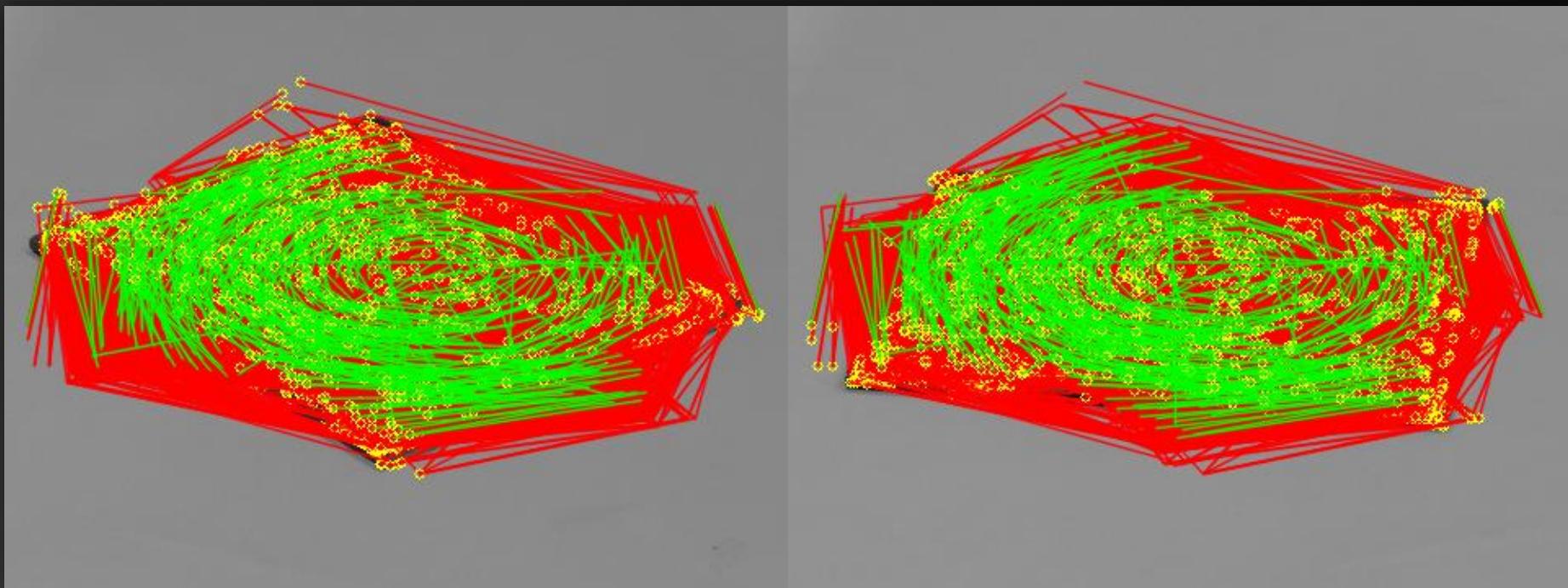
Matcher

- ASIFT Matcher
 - Remove identical matches



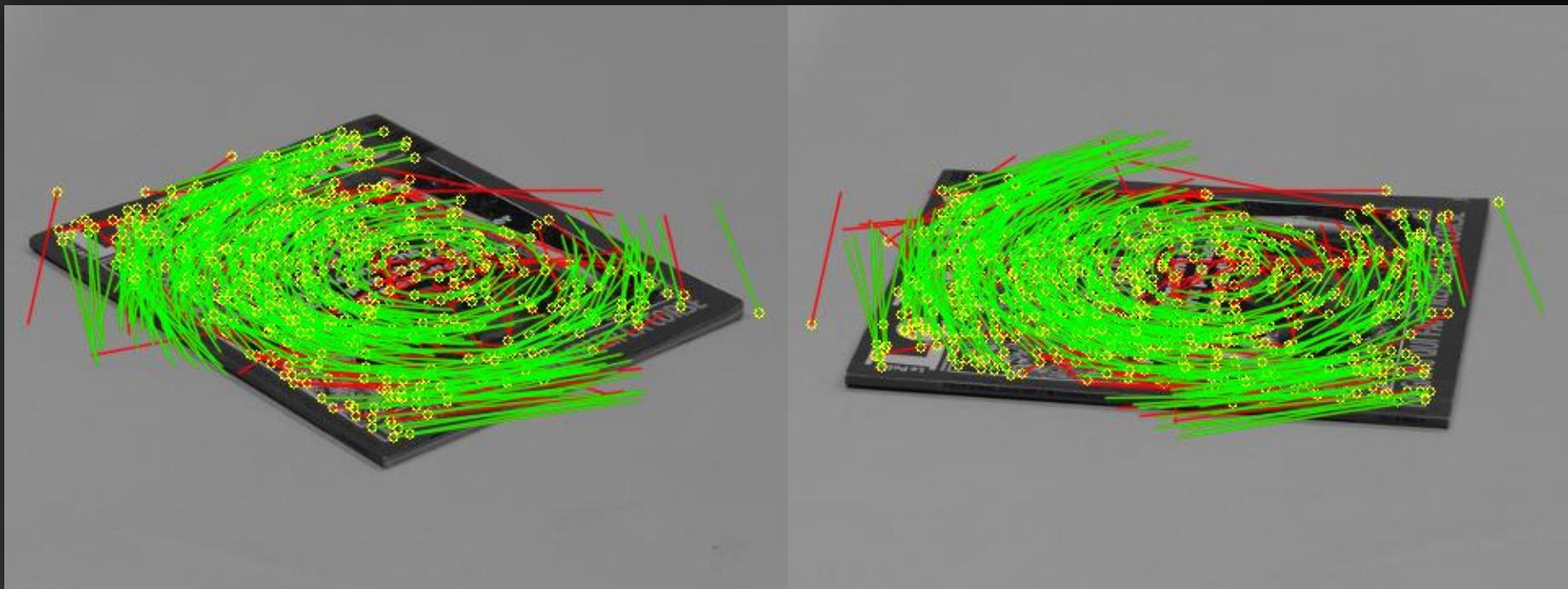
Matcher

- ASIFT Matcher
 - Remove ambiguous matches



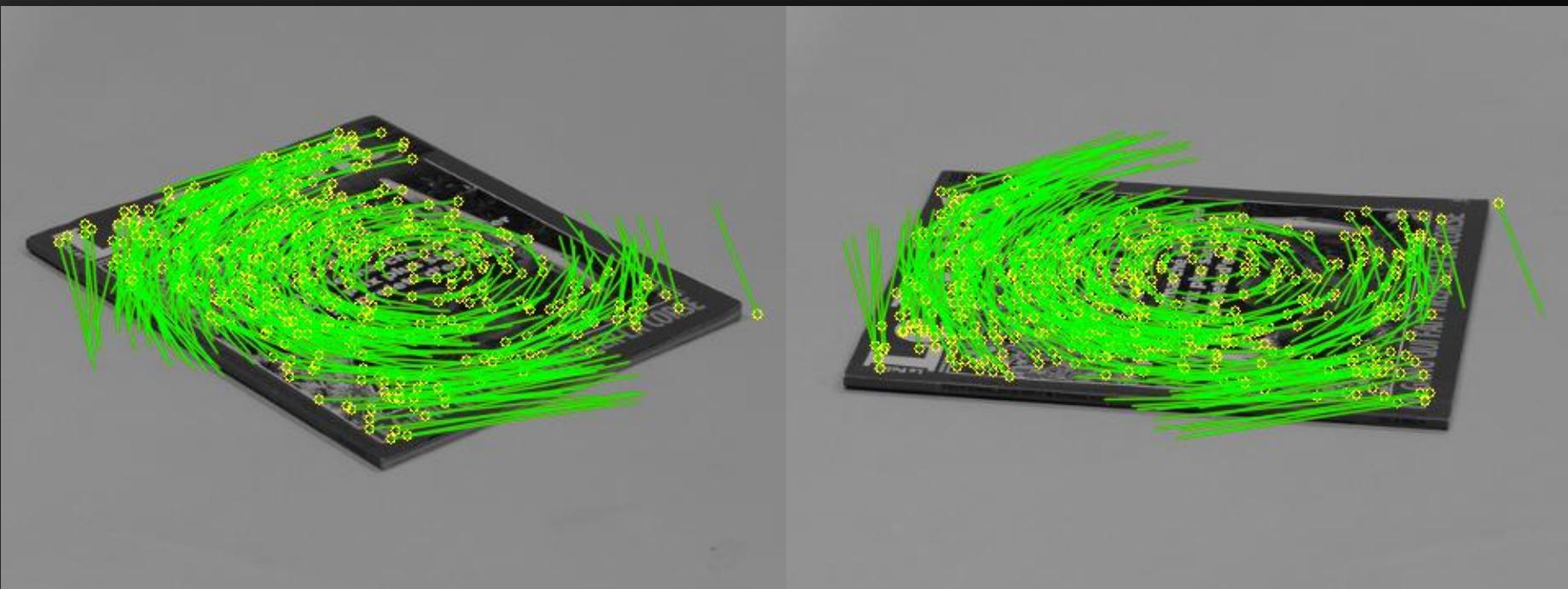
Matcher

- ASIFT Matcher
 - Filter epipolar inliner (RANSAC)



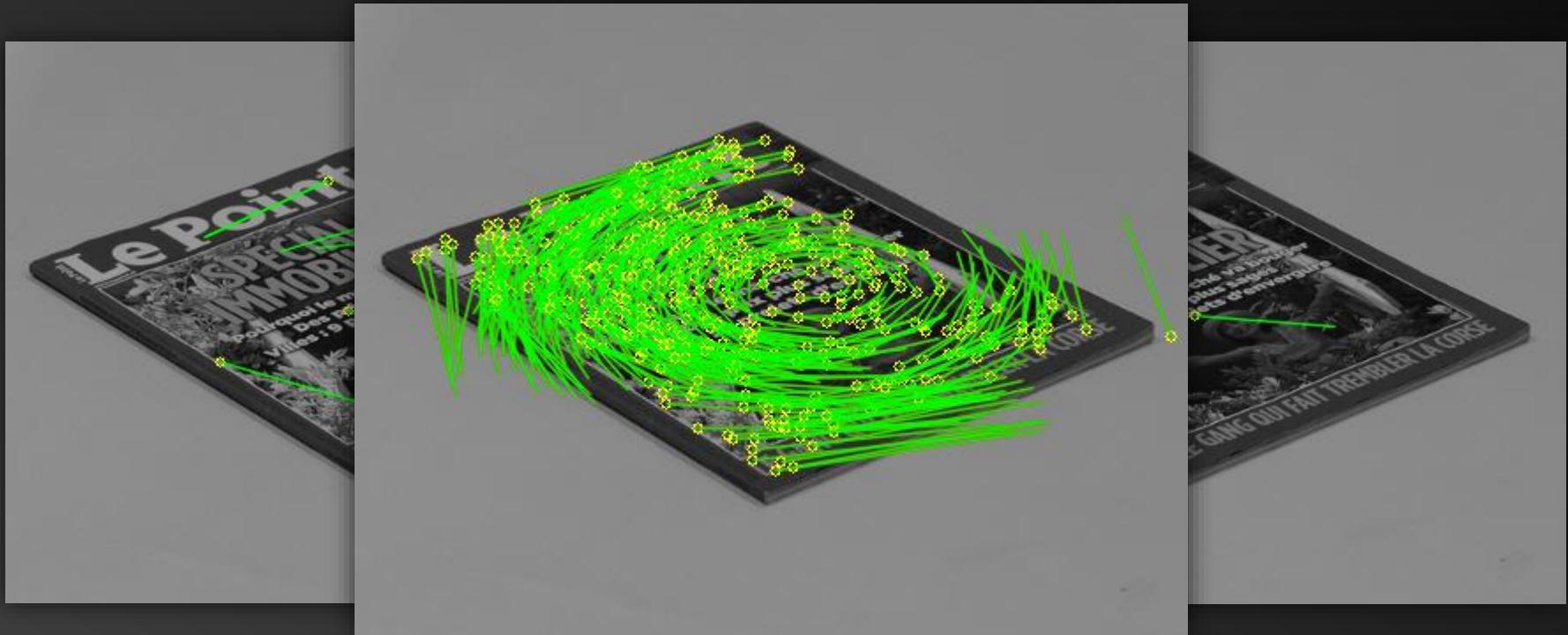
Matcher

- ASIFT Matcher
 - Final Matching



Matcher

- ASIFT Matcher
 - Robust and Epipolar Matcher results

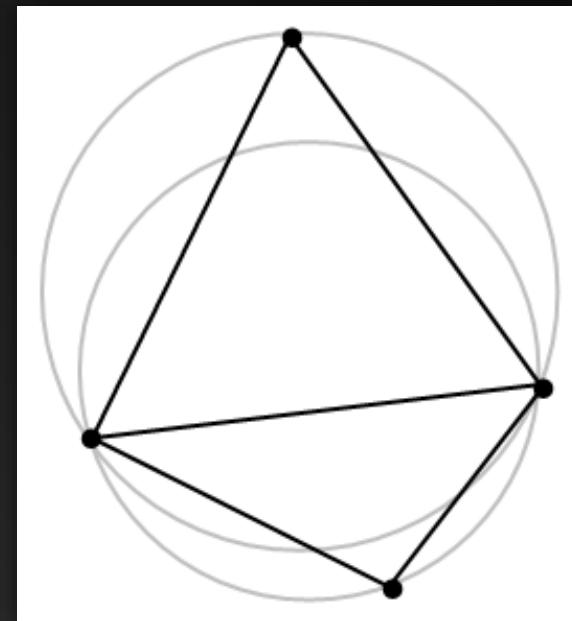
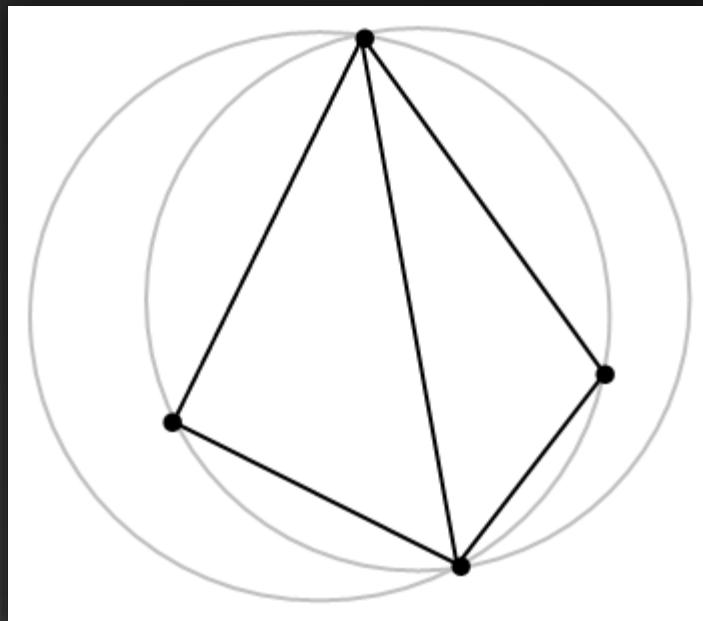


Spatial Interpolation

3) Triangulation

Triangulation

- Delaunay Triangulation



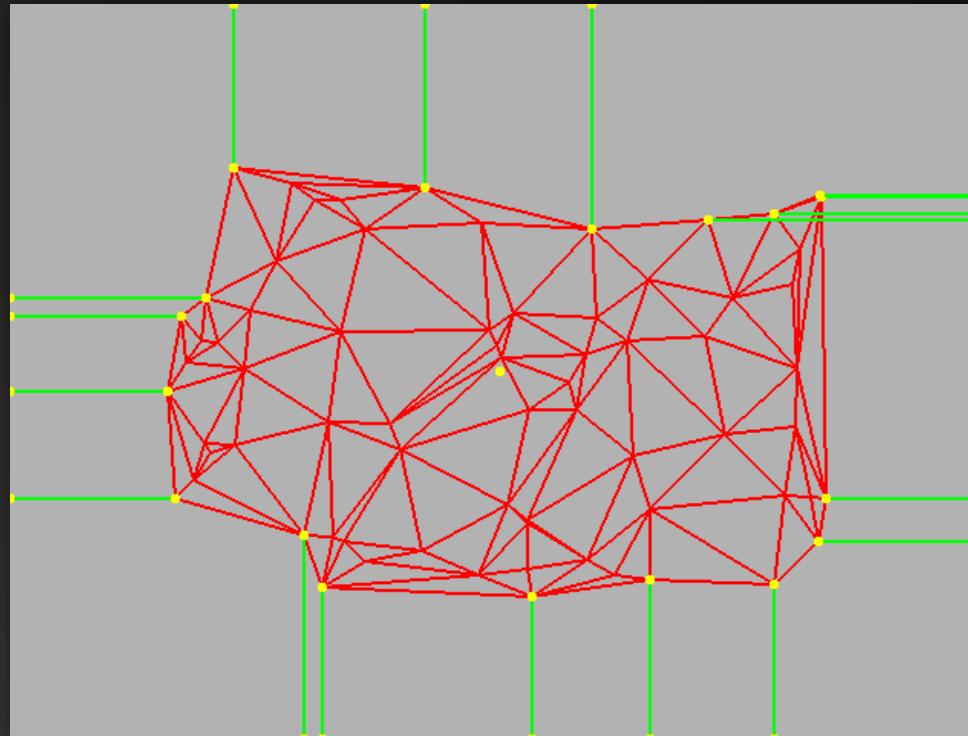
Triangulation

- Delaunay Triangulation



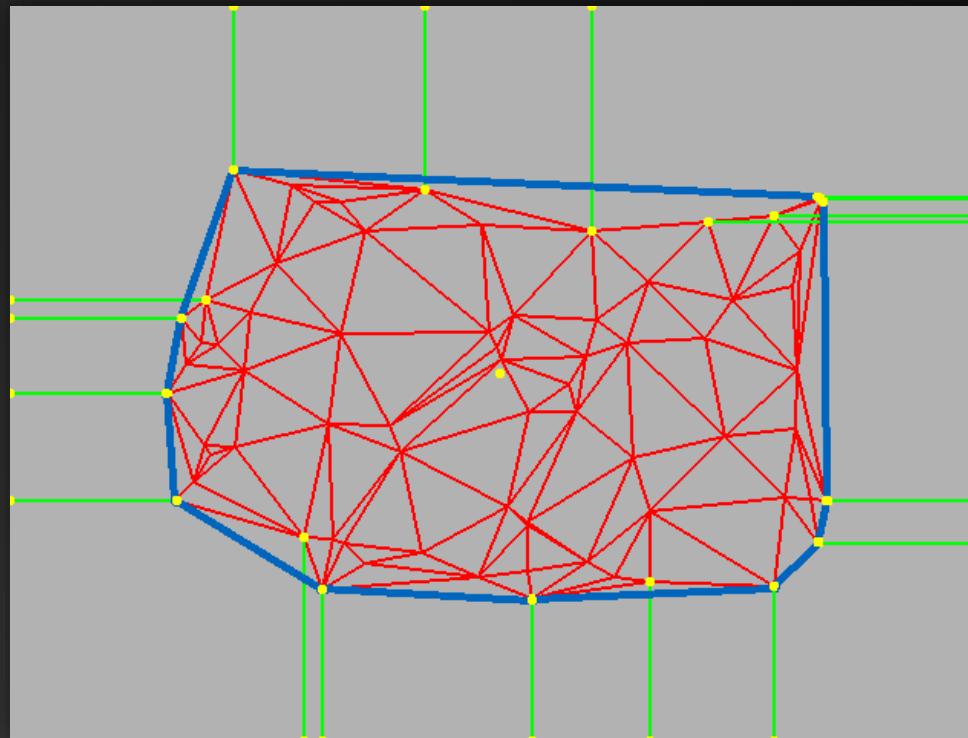
Triangulation

- Border Extrapolation
 - Orthogonal projection



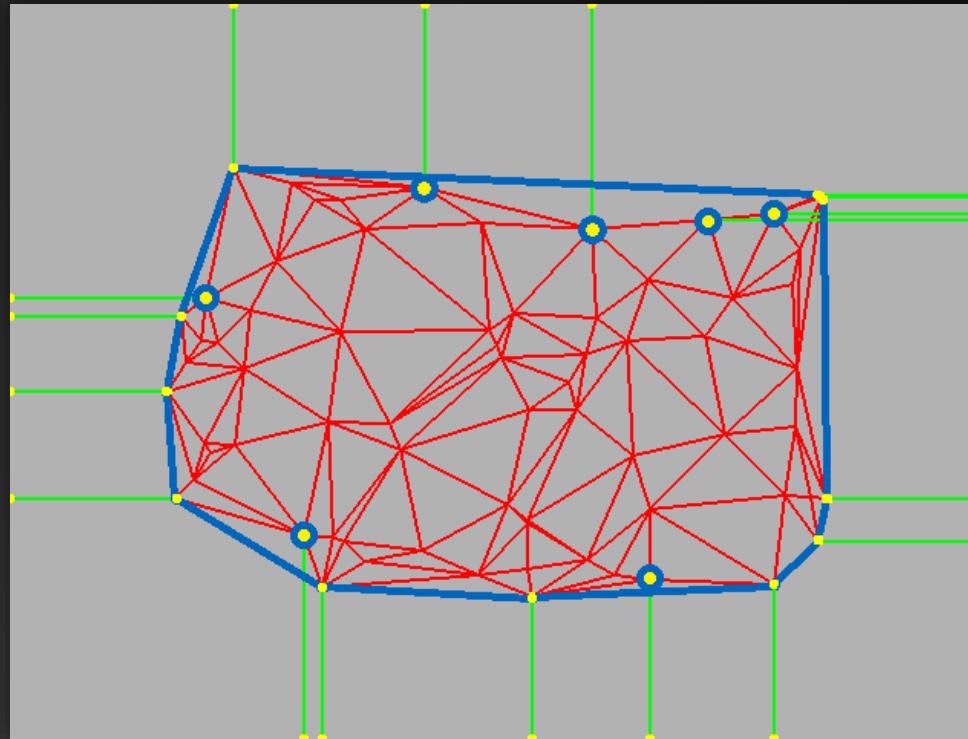
Triangulation

- Border Extrapolation
 - Convex hull



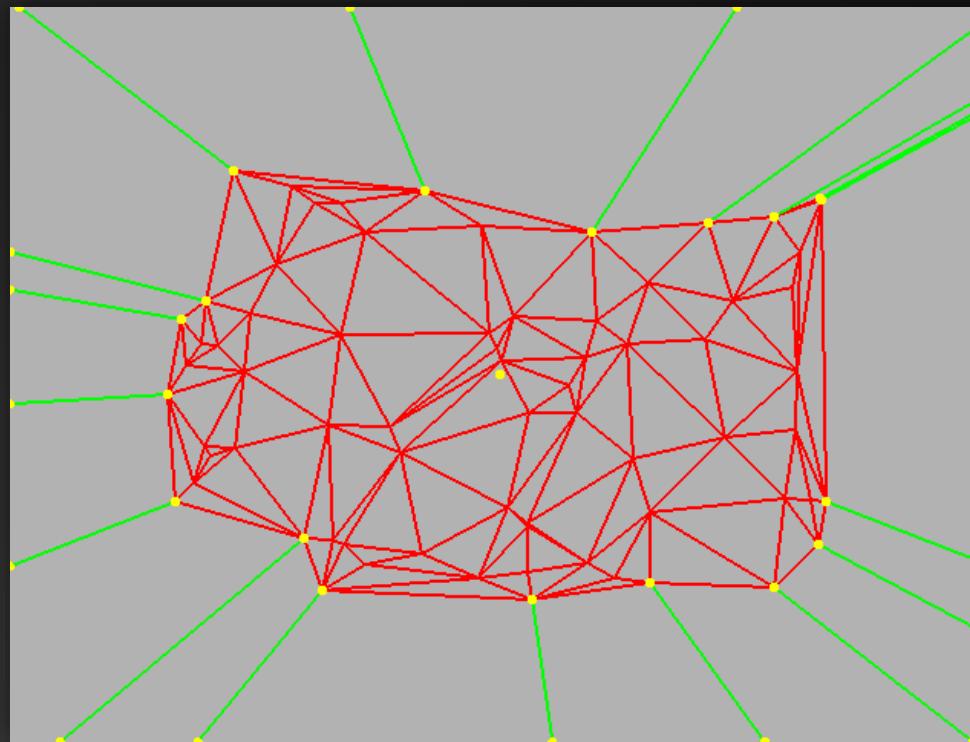
Triangulation

- Border Extrapolation
 - Left out points



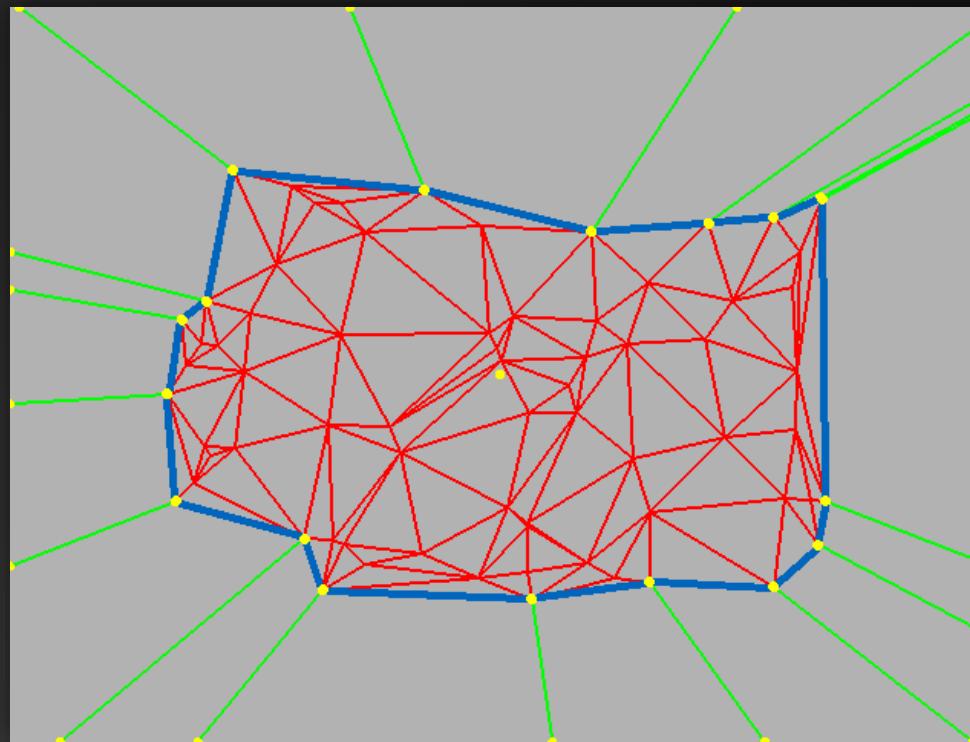
Triangulation

- Border Extrapolation
 - Radial projection



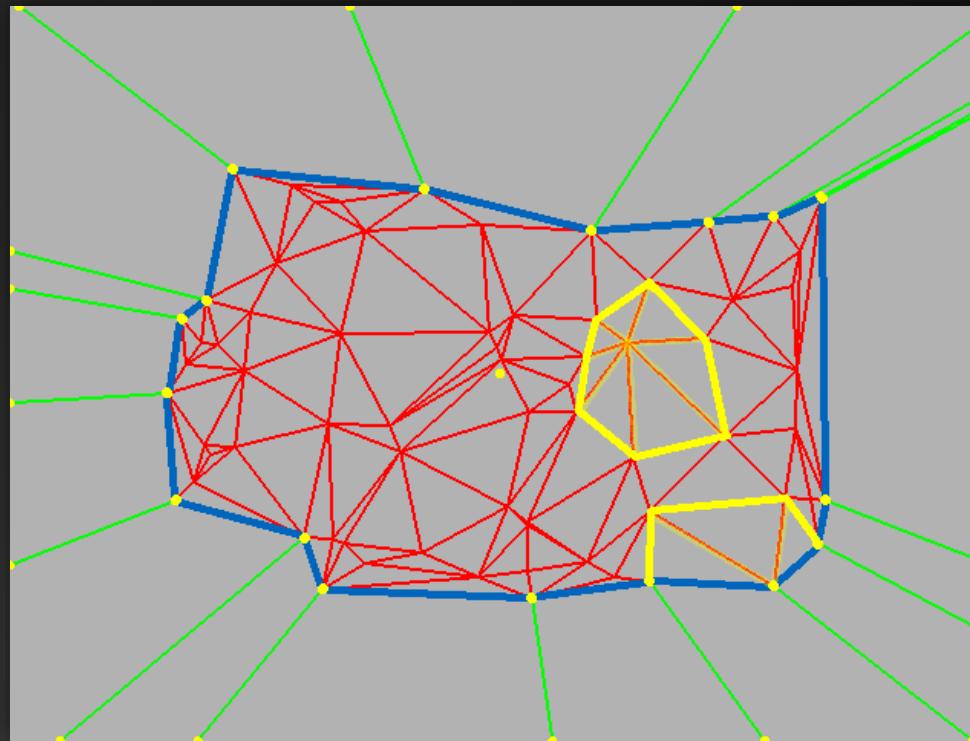
Triangulation

- Border Extrapolation
 - Point set contour



Triangulation

- Border Extrapolation
 - Exemplary induced wheel graphs



Triangulation

- Delaunay Triangulation



Triangulation

- Delaunay Triangulation w. Border Extrapolation



Spatial Interpolation

4) View Morphing

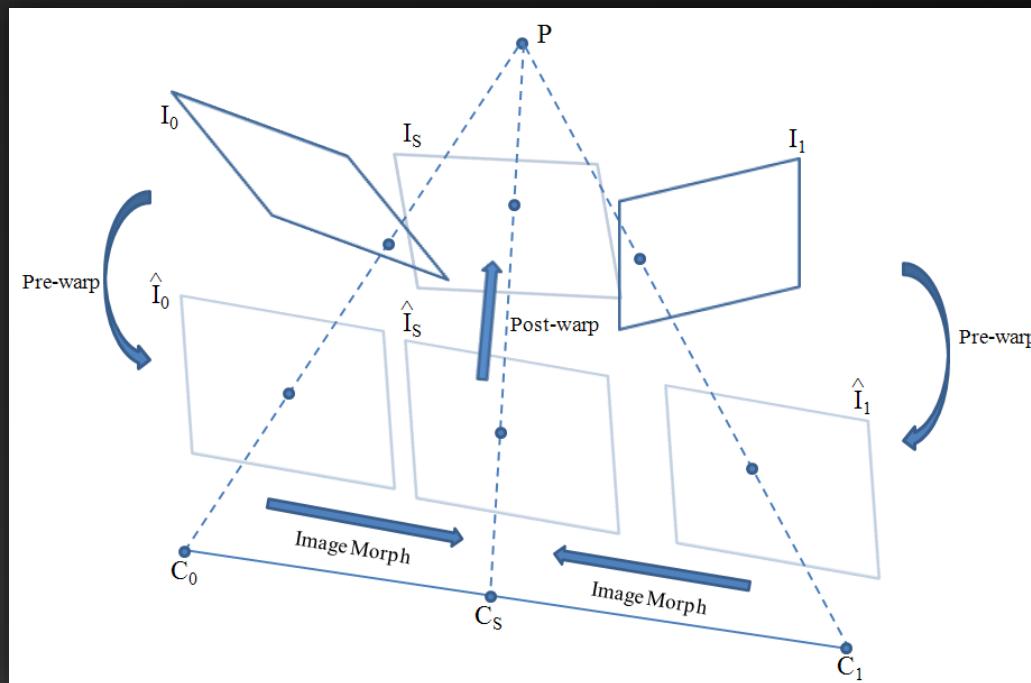
View Morphing

- Render textured triangle mesh
- Linearly interpolate triangle vertices from source to target
- Can result in unrealistic transformations!

Demo: Gogh, w/o rectification

View Morphing

- Seitz and Dyer introduce View Morphing [SD96]
- Consists of pre- and postwarping



View Morphing

- Seitz and Dyer introduce View Morphing [SD96]
- Consists of pre- and postwarping
- Backprojection using homography interpolation, as introduced by Fragneto et al. [FFR+12]

Given two rectification homographies $\mathbf{H}_1, \mathbf{H}_2$. An intermediate homography at $t \in [0, 1]$ is given by

$$\mathbf{H}(t) = \mathbf{H}_1 \mathbf{H}^t = \mathbf{H}_1 \cdot \exp(t \cdot \log(\mathbf{H})) ,$$

where

$$\mathbf{H} = \mathbf{H}_1^{-1} \mathbf{H}_2 .$$

$$\mathbf{H} = \mathbf{H}^{\mathsf{T}} \mathbf{H}^{\mathsf{T}} .$$

Demo: Gogh, w/ rectification

Discussion

Discussion

- Presented autonomous, easy to use multi-view video interpolation framework
- Works great for slow motion effects and certain view changes
- Fails for large foreground/background motion
- Does not handle occlusions

Thank you!

Time for more results! 😊